


3	2	1	0
7	6	5	4
B	A	9	8
F	E	D	C
+	PC	DA	AD
	RS	ST	GO

# DE KENNERS 14

1	4	0	0	0	0
---	---	---	---	---	---

KIM GEËRUKERS CLUB NEDERLAND  
4de Jaargang Nr. 14  
20 december 1980


--

--

--



Samenstelling van het bestuur:

Voorzitter

: Co Filmer

Dorpsstr 1051

1566 JE ASSENDELFT

Tel.: 075 - 210023

: Anton Muller

Sint Semeynsstr 78 1

1061 GM AMSTERDAM

Tel.: 020 - 860245

: Ted Schouten

Junoplns 57

2024 RM HAARLEM

Tel.: 023 - 257171

Postgitorrek.nr.: 3757649

Regeling accommodatie voor

KIM-club bijeenkomsten

: Bob van de Oudewetering

Industriewg 12

2102 LM HEEMSTEDDE

Tel.: 023 - 286444

Technisch adviseur,

cassette programma bibliotheek

en propaganda KIM-club

: Uwe Schröder

Echternach 161

5625 KC EINDHOVEN

Tel.: 040 - 421821

: Sebo Woldringh

Kliverink 619

1104 KC AMSTERDAM ZUIDOOST

Tel.: 020 - 900085

Organisatie, hardware en

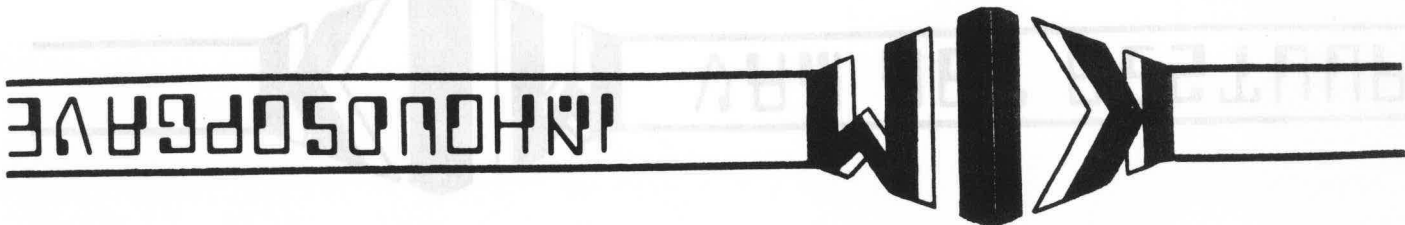
beheer KIM-club-KIM

: Rinus Vleesch Dubois

F Nijhtingalestr 212

2037 NG HAARLEM

Tel.: 023 - 330993



Pagina:

Inhoudsopgave	1
Inhoudsopgave	2
Van het bestuur	3
Van de Redactie	4
JUNIOR H J C Otten	8
One armed bandit for the junior	12
Ervaringen met de Junior (J van Sprang,	14
W L van Pelt)	16
JUNIOR Leestrouine voor Kansas City	18
cassette interface (Sebo Woldringh)	26
JUNIOR Schrijffrouine voor Kansas City	27
cassette interface (Sebo Woldringh)	36
PET Initiatisator (R Uphof)	39
System software: RAE printer bootstrap	40
KIM Schaakprogramma (T Kortekaas)	
Apple tapes met de KIM (F Messander en	
H J C Otten)	
Patches op Microsoft BASIC (HJC Otten)	
FLOPPY DISC AANWIJZINGEN	

De KIM KENNER is

een uitgave van

de KIM gebruikers

club Nederland.

Adres voor het in-

zenden van en re-

akties op artike-

len voor de KIM

KENNER:

p/a H.J.C. Otten

Dr Schaepmanstr 15

1581 BG WEESP

Tel.: 02940-13349

Redactie KIM KENNER:

Anton Muller

Hans Otten

Peter Visser

Geheel of gedeelte-

lijke overname van

de inhoud van de

KIM KENNER zonder

toestemming van

het bestuur is ver-

boden.

Toepassen van gepu-

bliceerde programma's,

hardware etc. is al-

leen voor persoonlijk

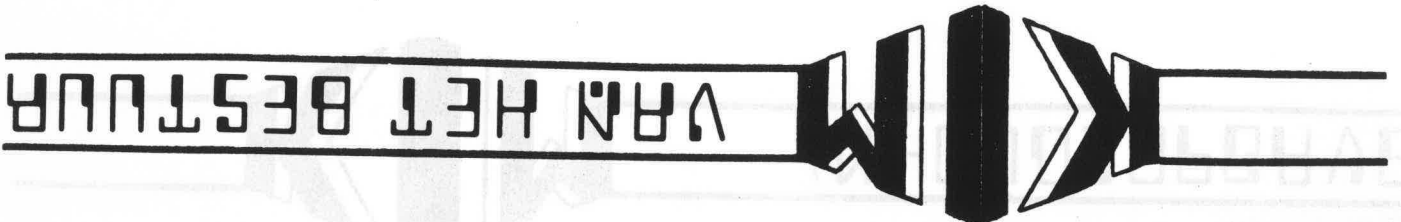
gebruik toegestaan.

(C) 1980 by KIM Gebru-

kers club Nederland.

Verschijnt vijfmaal

per jaar.



KIM - GEBRUIKERSCLUB

#### Forum

Op bepaalde bijeenkomsten wordt in het programma ruimte gereserveerd voor een forum.

#### Doel van een forum

Parate beschikbare kennis en informatie mee te geven in antwoord op de door de leden gestelde vragen. De leden kunnen tijdens het forum-gebeuren hun problemen, t.w. hard- en/of software de vrije loop laten. Tevens kunnen ideeën en voorstellen gelanceerd worden. Voorddeel van deze directe communicatie is praktische oplossingen voor de vragensteller, zodat deze sneller vooruit komt. Wederom is gebreken dat er binnen de KIM-club veel kennis aanwezig is (ook voor andere 6502 systemen).

Een levendige discussie kwam op gang bij KIM compatible JUNIOR, waarbij forumleden Hans en Anton onze leden volstoppen met informatie om KIMprogramma's op de junior te kunnen draaien.

Samenstelling forum september was:

Hans Otten - Anton Muller - Rinus Vleesch Dubois - Wim van Gelderen - Uwe Schröder - Sebo Woldringh - Co Filmer.

#### KIM-MARKT VOOR U BEKEKEN

Zoals gewoonlijk wordt er op elke bijeenkomst enige tijd voor de "MARKT" ingeruimd. T.a.v. de normale gang van zaken en spelregels: 1. Aanmelden dient voor de aanvang van de "MARKT" te geschieden, zodat een ieder op volgorde van aanmelding aan de beurt komt. 2. Afhankelijk van het aantal personen die iets hebben aan te bieden, wordt de beschikbare tijd verdeeld in "x" minuten per persoon. 3. Alles wat met micro's, terminals, onderdelen, boeken, tijdschriften, software, of wat hiermede in verband gebracht kan worden (in de ruimste zin van het woord), kan aangeboden worden. 4. De KIM-club cq. Bestuur kan niet verantwoordelijk of aansprakelijk gesteld worden voor o.a.:

- Goede werking van het aangeboden goed.
- Defecten en/of garantie.
- Plaats van herkomst en eventuele gevolgen.
- Afspraken betreffende leveringen, levertijden, betalingen, enz.

Globale opsomming van het gebodene tijdens "meeting september te Heemstede"

Hr. v.d. Burg

Teletype linten

Hr. Visser

Prof. 6502 systemen/printen voor videoterminals/8K

Hr. Meijer

Nummerdisplays 0-9 lcl latch-kims + dokumentatie

Hr. Schröder

Boeken, OSI board, Applecomputers, OKI printers, Macro-

Hr. Verkooy

Printers (gebruikt)

assembler, dynamic ramboards



H.J.C. Otten

Dit is het vijfde en laatste nummer van de KIM KENNER in 1980. Dit nummer ontvangt U in de maand vol feestdagen december, namens redactie en bestuur wens ik U prettigfeestdagen en de beste wensen voor 1980. Als we zoals gebruikelijk in deze tijd terugkijken op het afgelopen jaar, dan is dat voor de KIM 6502 club een goed jaar geweest. In 1980 is de KIM club uitgegroeid tot een echte 6502 gebruikersclub met een flinke groei in ledental. We hebben vele nieuwe enthousiaste leden met allerlei 6502 systemen. Ook in 1981 zal de KIM club vele activiteiten kunnen ontwikkelen, waarbij ook uw bijdrage welkom en nodig is. Een club als de KIM club wordt geheel gedragen door de leden, waarbij de bijdragen van gevorderden als beginners nodig zijn. Als we allemaal ons best doen dan kunnen we onze interesse, de computer, door gezamenlijke inspanning voor ieder inhoud geven.

Een van de belangrijkste activiteiten van de KIM club is de tweemaandelijksse bijeenkomst. Een belangrijk onderwerp van de bijeenkomst op 17 januari zal de taal FORTH en de implementatie daarvan op allerlei 6502 systemen zijn. We zijn er erg blij mee dat we een bekend deskundige, Dr. Hans Nieuwenhuizen, secretaris van de Europese FORTH gebruikers Groep, bereid hebben gevonden daar een ongetwijfeld leerzame en interessante inleiding in de vorm van een lezing over te willen houden. Na deze voor ieder te volgen lezing zal voor de meer geïnteresseerden een workshop onder leiding van Dr. H. Nieuwenhuizen worden gehouden. In ieder geval kan dan met FORTH op Apple's met een diskdrive worden gewerkt, neem uw Apple gerust mee daarvoor. Later kan FORTH op allerlei 6502 systemen draaien.

FORTH is een hogere programmeertaal met uitstekende eigenschappen voor real time controle toepassingen, en stelt daarbij beschikbare eisen aan intern en extern geheugen. Ik raad U aan van te voren Byte van augustus 1980 en met name het artikel 'What is FORTH?' te lezen en ook de Byte's van september en oktober over FORTH. Het voordeel van een onafhankelijke club zoals de KIM club is dat we samen sterk staan. Laat daarom uw positieve of negatieve ervaringen met aankoop van hard- en software weten aan uw medeclubleden.

De gelijkenis tussen de KIM en Junior hard- en software is dusdanig groot, dat het zonder gecompliceerde programma - wijzigingen mogelijk is software van de ene aan de andere computer aan te passen. Daarbij is echter documentatie nodig van die andere computer.

KIM bezitters (en ook andere 6502 gebruikers) zou ik aanraden de Junior boeken te kopen, vooral deel 2 met de Junior monitor is goed leesbare en interessante literatuur over de 6502 en natuur-  
lijk de Junior.

Mijn eerste indruk van de Junior was dat het een kopie van de KIM was. Wat betreft de hardware is dit ook zonder meer waar, er is één detail waarin de Junior en de KIM duidelijk verschillen en dat is de wijze van aansturen van de LED display's. Het lijkt precies op elkaar, alleen de polariteit is omgedraaid. Als we bij de KIM een één naar een LBD-segment sturen, dan moet bij de Junior een nul worden gezonden. Verder is de werking identiek.

Dit verschil uit zich duidelijk in de hex-naar zevenssegment tabel, de tabel in de Junior (LF0F en verder) is gelijk aan die van de KIM (LF07 en verder) maar bit voor bit geïnverteerd!

Junior gebruikers kunnen een grote hoeveelheid programma's vinden in het "First book of KIM", maar stuiten dan op onbekende KIM routines, waarvoor ze de KIM manuals moeten raadplegen.

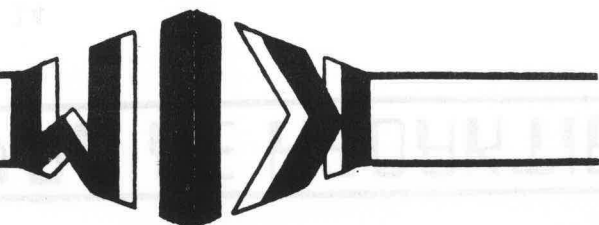
Om de conversie te ondersteunen zal ik een aantal overeenkomstige KIM en Junior routines aanwijzen die veel gebruikt zullen worden. Hieruit blijkt dat de software ook grote overeenkomsten vertoont, een aantal subroutines heeft dezelfde naam en coding. Er zijn echter een aantal opmerkelijke verschillen in de software.

De KIM heeft meer I/O mogelijkheden zoals een teletype en een audio cassette recorder, voorzieningen die bij de Junior ongetwijfeld zullen komen. Een raadgeving: wacht op de ontwikkelingen bij Elektron en ga niet zelf ontwikkelen of kopen, daar krijgt u spijt van. Doe zoveel mogelijk gemeenschappelijk!

H.J.C. Otten

Conversie van KIM naar JUNIOR en omgekeerd

JUNIOR







De KIM heeft RAM vanaf adres 0000 t/m 03FF evenlas de Junior .  
Daarnaast heeft de KIM RAM ( van de 6530's ) van 1780 t/m 17FF .  
De Junior heeft RAM van 1A00 t/m 1A7F , ook 128 bytes

## RAM Lokaties

Interrupt vector is bij de KIM de RAM lokatie 17FE en 17FF ,  
bij de Junior 1A7E en 1A7F .  
Non maskable vector is bij de KIM 17FA en 17FB , bij de Junior  
is dit 1A7A en 1A7B .  
De break vector ( instructie BRK ) is bij de KIM de interrupt  
vector 17FE etc , bij de Junior 1A7E etc.

## Vectoren

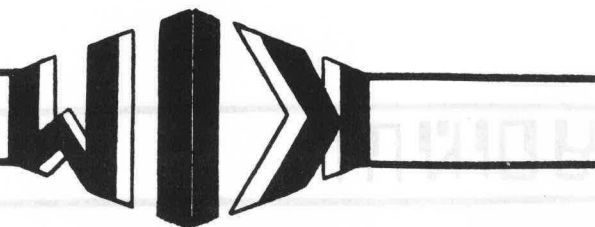
De 6530 en 6532 hebben gelijke timers , hiervoor is het beste de  
voor ledere serieuze 6502 gebruiker onontbeerlijke Program en  
HardwareManuals van de fabrikant te raadplegen .  
Timerlokaties 1704 etc van de KIM ontbreken bij de Junior .  
Overeenkomstige timers zijn de KIM lokaties 1744 t/m 174F en  
de Junior lokaties 1AF4 t/m 1AFF ( bv CN7A Junior 1AF4 = KIM  
C1K1F 1744 etc ) .  
De edge detect mogelijkheid van de Junior ontbreekt bij de KIM .

## Timers

Bij de KIM hebben we twee PIA's in de twee 6530 IC's . In de  
Junior zit er een in de 6532 . Ze gedragen zich gelijk , die 6530  
en 6532 als PIA , maar ze zitten op andere lokaties.  
KIM PIA 1700-1703 ontbreekt bij de Junior .  
KIM PIA 1740-1743 = Junior PIA 1A80-1A83 , vrijwel gelijk gebruikt.  
PAD KIM is bijvoorbeeld PAD Junior : 1740 = 1A80 .

## PIA Lokaties

JUNIOR





• • Laat het dan aan de KIM KENNER weten voor uw medegebruikers

RTS 1997-1998

Junior boek 2 van Elektuur : ( in assembler notatie : )

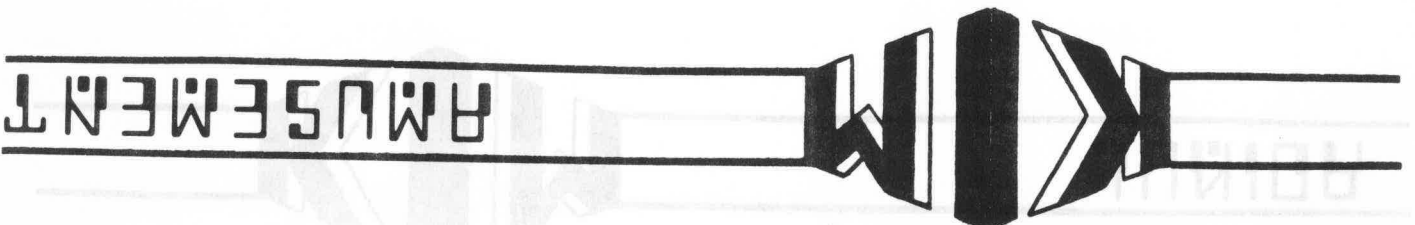
ԿՈՒՆՐԴ

```

0010 *****
0020
0030
0040
0050
0060
0070
0080
0090
0100
0110
0120
0130
0140
0150
0160
0170
0180
0190
0200
0210
0220
0230
0240
0250
0260
0270
0280
0290
0300
0310
0320
0330
0340
0350
0360
0370
0380
0390
0400
0410
0420
0430
0440
0450
0460
0470
0480
0490
0500

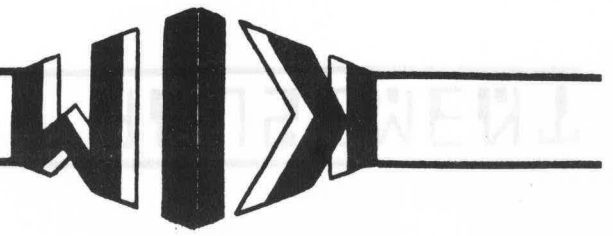
```

ONE ARMED BANDIT FOR THE JUNIOR MICRO COMPUTER  
 -----  
 ORIGINALLY A KIM-1 VERSION, MODIFIED BY  
 ANTON MUELLER  
 SINJ SEMINSTR 78 1  
 1061 GM AMSTERDAM  
 THE NETHERLANDS  
 IN ORDER TO MAKE IT SUITABLE FOR THE JUNIOR  
 THE PROGRAM INTERFACES WITH THE JUNIOR KEY-  
 BOARD AND DISPLAYS TO PRODUCE A ONE  
 ARMED BANDIT TYPE OF SLOT MACHINE ACTIVITY.  
 THE RIGHT-HAND SIDE OF THE DISPLAYS SHOWS  
 THE PLAYER'S BALANCE IN DUTCH FLORINS.  
 AN EARLIER VERSION OF THIS PROGRAM ALLOWS  
 THE WHEELS TO SPIN UNTIL THEY ARE STOPPED,  
 ONE AT A TIME, BY THE PLAYER TOUCHING A KEY.  
 WHILE THE EFFECT WAS GOOD, THEY LED TO  
 EXTREMELY HEAVY USAGE OF THE PUSHBUTTONS.  
 THIS VERSION CAUSES THE WHEELS TO STOP AUTO-  
 MATICALLY.  
 MUCH OF THE PROGRAM IS DEVOTED TO PRODUCING  
 THE FEEL OF THE SIMULATED MACHINE. IT  
 WOULD BE VERY EASY, OF COURSE, TO HAVE THE  
 WHEELS START AND STOP INSTANTANEOUSLY, AND  
 THE CASH BALANCE DISPLAYED IMMEDIATELY.  
 BUT THIS WOULD NOT RESEMBLE THE REAL  
 MECHANICAL MACHINE.  
 THERE IS NO HOUSE PERCENTAGE, SO THE  
 PLAYER CAN GO FOR CONSIDERABLE TIME ON HIS  
 INITIAL 25 GUILDERS. HOWEVER, HE IS NOT ALLOWED  
 TO WIN A TOTAL OVER 99 GUILDERS, AND THE  
 WHEELS WILL NOT SPIN IF HE LOSES ALL HIS  
 MONEY.  
 THE PROGRAM STARTS AT LOCATION 0200.  
 SPIN THE WHEELS BY TOUCHING ANY BUTTON  
 BETWEEN 0 AND F.  
 F. J. BUTTERFIELD  
 14 BROOKLYN AVENUE  
 TORONTO - ONTARIO M4M2X5  
 C A N A D A  
 -----



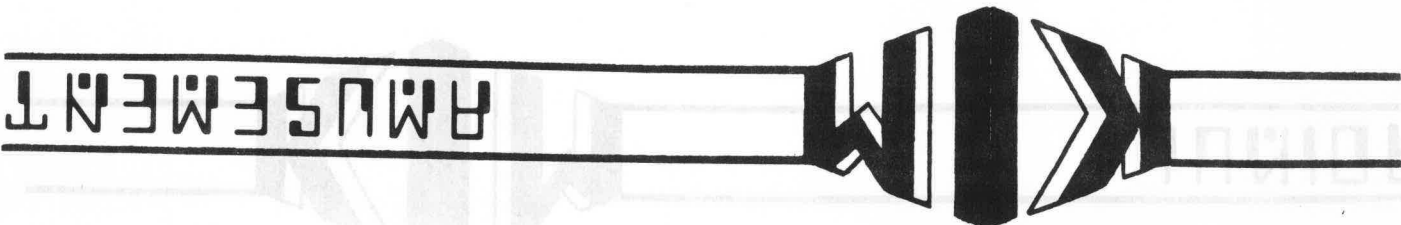


Address	Instruction	Comment
0510	*	WORK AREA'S IN PAGE ZERO
0520	*	
0530	BANDIT ORG	\$0200
0540	WINDOW *	\$0000
0550	ANT	WINDOW +05
0560	ARROW	ANT
0570	RWD	ARROW +01
0580	STALL	RWD +01
0590	TUMBLE	STALL +01
0600	*	
0610	*	REFERENCES TO JUNIOR MONITOR
0620	*	
0630	SCDSB *	*IDAC IS KEY DEPRESSED?
0640	FADD	*\$1A81
0650	FAD	*\$1A80
0660	FBD	*\$1A82
0670	LOOK	*\$1F0F
0680	*	
0690	*	MAIN PROGRAM STARTS HERE
0700	*	
0710	LDAM #25	GO
0720	STA	LDAM #25
0730	USR	ANT
0740	CVANT	STARTS WITH 25 BUCKS
0750	LDAM #00	CHANGE TO DISPLAY
0760	STA	ARROW
0770	MAIN DISPLAY LOOP	
0780	USR	DISPLAY
0790	END	LFA
0800	INC	TUMBLE
0810	USR	DISPLAY
0820	BEQ	ROLL
0830	LDAM #03	PAY 2 BUCKS
0840	STA	ARROW
0850	SEC	
0860	LDAM #01	CHARGE A BUCK
0870	STA	ANT
0880	CVANT	
0890	USR	ANT
0900	ROL	TUMBLE
0910	USR	DISPLAY
0920	DEC	STALL
0930	BNE	LFB
0940	LDX	ARROW
0950	LDAM #F7	
0960	ANDIM #F7	SPIN RESULT
0970	STAX	WINDOW +01 TO DISPLAY
0980	LSR	TUMBLE
0990	LSR	TUMBLE



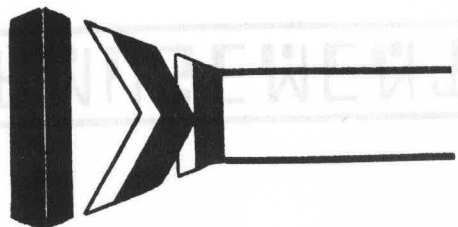
10100	023D	C6	06	DEC	ARROW	
10200	023F	D0	E7	BNE	LFB	
10300				*		ALL WHEELS STOPPED - COMPUTE PAYOUT
10400	0241	A5	04	LDA	WINDOW +04	
10500	0243	C5	03	CMP	WINDOW +03	
10600	0245	D0	37	BNE	NOMAT	
10700	0247	C5	02	CMP	WINDOW +02	
10800	0249	D0	33	BNE	NOMAT	
10900	024B	A2	10	LDXIM	#10	
11000	024D	C9	BF	CMPIM	#BF	PAY 15 BUCKS IF 3 BARS
11100	024F	F0	0D	BEG	PAY	
11200	0251	A2	0B	LDXIM	#0B	
11300	0253	C9	BD	CMPIM	#BD	PAY 10 BUCKS IF 3 UPS
11400	0255	F0	07	BEG	PAY	
11500	0257	A2	06	LDXIM	#06	
11600	0259	C9	BB	CMPIM	#BB	PAY 5 BUCKS IF 3 DOWNS
11700	025B	F0	01	BEG	PAY	
11800	025D	CA		DEX		
11900				*		A MIN. PAY AMOUNT IN X
12000	025E	86	07	PAY	STX	
12100	0260	A9	80	PAY	RMD	
12200	0262	80	08	STA	STALL	
12300	0264	Z0	8D	LFC	DISPLY	
12400	0267	C6	06	DEC	STALL	
12500	0269	D0	F4	BNE	LFC	
12600	026B	C6	07	DEC	RMD	
12700	026D	F0	9C	BEG	LFA	
12800	026F	18		CLO		
12900	0270	F8		SED		
13000	0271	A0	0C	LDA	AMT	
13100	0273	69	01	ADCM	#01	
13200	0275	B0	94	BOS	LFA	
13300	0277	80	05	STA	AMT	
13400	0279	Z0	BA	02	USR	CVAMT
13500	027C	D0	E2	BNE	FAX	
13600				*		WHEELS NOT ALL SAME - CHECK FOR SMALL WIN.
13700	027E	A2	03	NOMAT	LDXIM #03	PAY 2 BUCKS
13800	0280	C9	B9	CMPIM	#B9	IF CHERRY
13900	0282	F0	DA	BEG	PAY	
14000	0284	Z0	8D	02	02	LCK
14100	0287	A0	05	LDA	AMT	
14200	0289	D0	80	BNE	LFA	
14300	028B	F0	F7	BEG	LCK	
14400				*		DISPLAY SUBROUTINE
14500				*		
14600	028D	A6	06	DISPLY	LDX	ARROW
14700	028F	10	02	BPL	INDIS	
14800	0291	F6	02	OVER	INCA	WINDOW +02
14900	0293	CA		DEX		
15000	0294	10	FB	BPL	OVER	





AMUSEMENT

1510	0296	A9	7F	LDAIM	#7F
1520	0298	8D	81	STA	PADD
1530	029B	A0	0B	LDIYM	#0B
1540	029D	A2	04	LDXIM	#04
1550	029F	B0	00	LDAAX	WINDOW
1560	02A1	8C	82	STY	FBD
1570	02A4	8D	80	STA	PAD
1580	02A7	D8		CLD	
1590	02A8	A9	7F	LDAIM	#7F
1600	02AA	E9	01	SECIM	#01
1610	02AC	D0	FC	BNE	ZIP
1620	02AE	8D	82	STA	FBD
1630	02B1	C8		INY	
1640	02B2	C8		INY	
1650	02B3	C8		DEX	
1660	02B4	10	E9	BPL	LITE
1670	02B6	20	AC	JSR	SCDSB
1680	02B9	60		RIS	
1690					
1700					AMOUNT CONVERSION
1710					
1720	02BA	A0	00	LDA	AMT
1730	02BC	29	0F	BNDIM	#0F
1740	02BE	AA		TAX	
1750	02BF	BD	0F	LDAAX	LOOK
1760	02C2	80	00	STA	WINDOW
1770	02C4	A0	00	LDA	AMT
1780	02C6	4A		LSRA	
1790	02C7	4A		LSRA	
1800	02C8	4A		LSRA	
1810	02C9	4A		LSRA	
1820	02CA	AA		TAX	
1830	02CB	BD	0F	LDAAX	LOOK
1840	02CE	80	01	STA	WINDOW+01
1850	02D0	60		RIS	
SYMBOL TABLE 3800 38A8					
AMT	0000	ARROW	0006	BANDIT	0200
DISPLY	028D	GO	0200	INDIS	0293
LOK	0284	LOOK	1F0F	LFA	020B
LFC	0264	NOMAT	027E	OVER	0291
PADD	1A81	PAD	1A80	FAY	0260
FBD	1A82	ROLL	0210	RMD	0007
STALL	0008	TUMBLE	0009	WINDOW	0000
AMT	02BA	CVANT	029A	LITE	029F
DISPLY	028D	GO	0200	INDIS	0293
LOK	0284	LOOK	1F0F	LFA	020B
LFC	0264	NOMAT	027E	OVER	0291
PADD	1A81	PAD	1A80	FAY	0260
FBD	1A82	ROLL	0210	RMD	0007
STALL	0008	TUMBLE	0009	WINDOW	0000



J. van Sprang

Tulp 71

Krimpen a.d. IJssel

W.L. van Pelt

Jacob Jordaanstr. 15

Krimpen a.d. IJssel

Ervaringen met de JUNIOR

De ervaringen, wensen en verzuchtingen die in een vorige KIM KENNEDY zijn vermeld vormen niteraard geen afgerond geheel. Vandaar deze

voortzetting.

Bemerkt is dat door ook deze publicatie contacten zijn ontstaan en

verstevigd. Inmiddels hebben bovengenoemde elkaar gevonden in een

gemeenschappelijk probleem: de behoefte om programma's op de band

te zetten, zodat een hoop toetswerk eenmalig kan blijven. Vanwaar

de behoefte, als er nog geen JUNIOR-programma's zijn verschenen?

Dat bleek niet zo'n nood, want via de sekretaris verkregen we in-

zicht in het verschil tussen KIM en JUNIOR monitor routines. Niet

dat daarmee alles kan worden vertaald in JUNIOR, maar een paar pro-

gramma's zijn nu in orde. Ook door contacten met andere leden elders

konden over en weer gegevens worden uitgewisseld. Wij hopen dat het

bestuur die verschillen, maar dan wel alle, tussen KIM en JUNIOR, in

de KIM KENNEDY zullen publiceren, zodat een ieder daar profiteert van

heeft. (Is reeds gebeurd. Zie artikel in dit nummer van Hans Otten;

Red.) Gezien onze behoefte werd, zodra de mogelijkheid daar was en

in weerwil van de raad die ons gegeven was, een cassette-interface

bij Musiprint Computer Products aangeschaft. (Deze firma is ook

bekend onder de naam CHIP. Red.). Deze interface leverde wel wat

stof op. Nu het uiteindelijk werkt, zeg je dat de interface op zich

in orde is. Maar om het ding echt aan de praat te krijgen hebben

begijnelingen wel een helpend handje nodig in de vorm van een pro-

gramma. Daar had MCP voor gezorgd, zoals ze al eerder inzag dat de

concurrentie toch van je vergt dat de JUNIOR computer beter verkocht

kan worden met bedrukte/gegraveerde toetsen dan met die afgetijp-

ke plakcijfers en -letters en zoals je beter kunt aanrijzen met

gratis programma's. Niet al die programma's zijn uitgetroefd, maar

zeker is, dat ze niet allemaal werken. Voor wat het cassette-interface

programma betreft: er is vele malen telefonisch contact gezocht met

MCP en als je iemand kreeg met meer gezag dan de telefoniste, dan

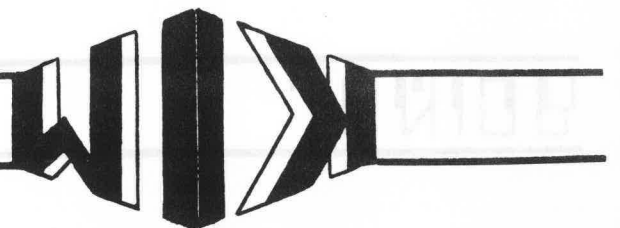
nog blijft dat we nu nog op een beter programma wachten. Wat we heb-

ben ontvangen bij de interface bleek goede voeding voor de printen-

mand. Uiteindelijk heeft de software-deskundige van de KIM club,

Sebo Woldringh, ons binnen een kwartier uit de nood geholpen met

nieuwe programmatuur voor de cassette in- en uitvoer. Daar moest

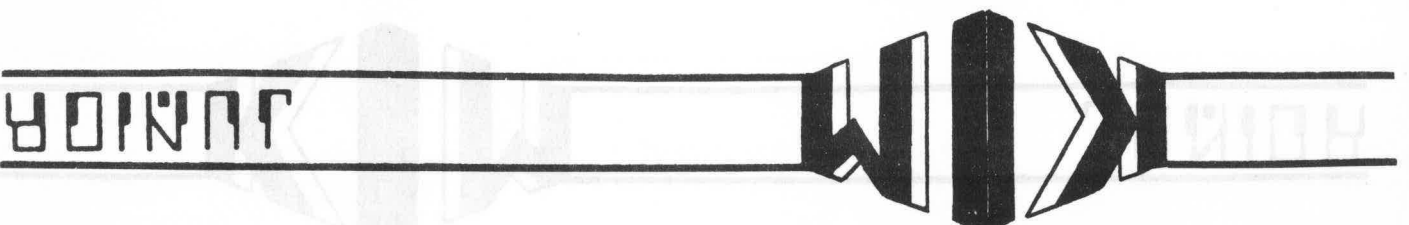


N.B.: De 2½ bit tijd op locatie 0148 blijkt soms beter te werken als je die op 24 (hexadecimaal 18) zet.

nog wel een kleinigheid aan veranderd worden, maar dat was in twee avonden rond. Dat ging dus duidelijk sneller dan bij MCP. De programma's vindt u hierbij. Nog even wat over de interface. Als u niet in het bezit bent van dit ding: over een paar maanden komt Elektuur met een cassette interface die veel meer belooft. Mocht u de MCP cassette interface toch willen kopen of al bezitten, dan zult u bemerken dat één der IC's is afgeschraapt en in de beschrijving een "Speciaal IC volgens fabrieksspecificaties" heeft. Wat er voor speciaals is aan een doorgewone 74123 flip-flop is ons niet bekend. De interface werkt met TTL-IC's waarvan bekend is, althans zo heb ik begrepen, dat deze een spanning van 2,4 Volt en hoger leest als "1" en 0,8 Volt en lager als "0". Kijkt u naar de technische gegevens, dan blijkt dat de audio-input maar liefst 1,5 Watt nodig heeft. Het atregelen van de interface kon gelukkig bij Van Sprang gebeuren die een praktische ervaring heeft met elektronica. Hij beschikte over een draagbare radio/cassette combinatie. Enige modificaties hieraan en klaar is Kees. Bij mij ging het wat anders. Aangemoedigd door een artikel van Hans Otten in Radio Bulletin nr. 3 van 1980, kocht ik bij Radio Service Twenthe in Den Haag de aanbding-inbouw-cassette-deck type Touring 108 ad f 32,50. Het ding wordt in een doos overhandigd en je gaat blij naar huis. Nu is het te hopen dat niet iedereen het overkomt dat bij thuis-komst, zoals bij mij, blijkt dat het meest essentiële onderdeel ontbreekt: de motor. Voor mij werd het door de niet geringe benzinekosten dus geen koopje. Enfin, de veding die Hans Otten vermeldde is rond gekomen. Hier bleek ook de noodzaak van samenwerking met anderen. Dat lag niet aan het schema, maar aan gebrek aan kennis en ervaring mijnerzijds. Je bent een leek, of niet. Ook de beide versterkers zijn gebouwd. Aangezien uiteindelijk gekozen is voor de universele luidspreker/versterker uit de Elektuur-Halgeleidergids 1980, hangt de eind-versterker van Hans Otten met 2' n 0,5 Watt nu aan het wachtrek. En zo ook de 1.10 maal versterker. Waarom dat? Wel, gesteld dat de gestandaardiseerde uitgangsspanning van recorders 200 mV is, zoals wij menen, dan bereiken we dus bij 10 maal zo'n 2 Volt. En dat is te weinig voor de TTL-IC's van onze interface, zoals hiervoor vermeld; De Elektuur-unit kan zo'n 200 maal versterken, dus daar waren dan ook geen problemen mee.

Nu we stad en land niet meer hoeven af te bellen, kunnen we eindelijk in deel 2 van het JUNIOR-boek duiken om daar te zien dat er inderdaad heel wat in de monitor zit. De behoefte aan uitbreiding zal na het lezen van dit deel wel glore.

- - -



READ

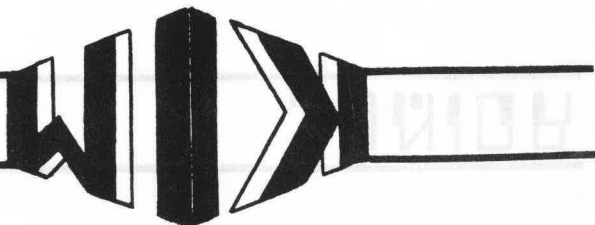
KIM SOFTWARE LIBRARY

PAGE 01

```
0010: 0100 READ ORG $0100
0020:
0030: *****
0040: *
0050: * LEES ROUTINE VOOR KANSAS CITY CASSETTE
0060: * INTERFACE VOOR DE JUNIOR MICRO COMPUTER
0070: * POOTJE 21 VAN DE APPLICATIE CONNECTOR
0080: * DOORVERBINDEN MET DE CASSETTE INPUT
0090: * START ADRES INTIKKEN OP SAL/SAH
0100: * EIND ADRES + 1 INTIKKEN OP EAL/EAH
0110: *
0120: * AUTEUR: SEBO WOLDRINGH
0130: * ----- KLIEVERINK 619
0140: * AMSTERDAM ZUIDOOST
0150: *
0160: * COPYRIGHT (C) BY S. T. WOLDRINGH
0170: *
0180: *****
0190:
0200:
0210:
0220: * HALF * $0005 (HALVE BIT TIJD = 5)
0230: * HEEL * $000A (HELE BIT TIJD = 10)
0240: * TWEEH * $0019 (TWEEN EN EEN HALVE BIT TIJD = 25)
0250:
0260: PAGE ZERO LOCATIES
0270:
0280: INL * $00F8
0290: INH *
0300:
0310: PIA POORT B VAN 6532
0320:
0330: PBD * $1A82 PIA B DATA REGISTER
0340: PBD * $0100 PIA B DATA DIRECTION REGISTER
0350:
0360: GEBRUIKTE TIMER LOCATIE:
0370:
0380: CNTD * $1AF7 1024 TIMER
0390:
0400:
0410:
0420:
0430:
0440: LDA SAL HAAL START ADRES LOW OP
0450: STA INL EN ZET DIE IN PAGE ZERO START ADR LOW
0460: LDA SAH HAAL START ADRES HIGH OP
0470: STA INH EN ZET DIE IN PAGE ZERO START ADR HIGH
0480: LDAIM $00 ZET DATA DIRECTION REGISTER B
0490: STA PBD OP INPUT
0500: READA JSR READI LEES EEN BIT
```



JUNIOR



```

0510: 0112 D0 FB BNE READA BLIJF Wachten tot er een "0" komt
0520: 0114 A9 05 LDAIM HALF ZET HALVE BIT TIJD
0530: 0116 20 4B 01 READB JSR EN HAAL EEN BIT OP
0540: 0119 D0 F4 BNE READA BLIJF Wachten tot er wat komt
0550: 011B A0 08 LDAIM #08 ZET AANTAL BITS PER BYTE
0560: 011D 48 PHA OP DE STACK
0570: 011E A9 0A READC LDAIM HEEL ZET HELE BIT TIJD
0580: 0120 20 4B 01 JSR EN HAAL EEN BIT OP
0590: 0123 18 CLC DE CARRY FLAG WORDT GEZET
0600: 0124 F0 01 BEQ READD ALS WE EEN "1" HEBBEN
0610: 0126 38 SEC GELEZEN
0620: 0127 68 PLA READD SCHUIF DE GELEZEN
0630: 0128 2A ROLA BITS IN DE ACCUMULATOR
0640: 0129 48 PHA
0650: 012A 88 DEY VERLAAG LOOP COUNTER
0660: 012B D0 F1 BNE READC
0670: 012D 68 PLA IN ACC STAAT NU EEN CHARACTER
0680: 012E 91 F8 STAIY INL DAT WE STOREN OP
0690: 0130 E6 F8 INC INL HET AANGEGEVEN
0700: 0132 D0 02 BNE READE GEINDEXEERDE
0710: 0134 E6 F9 INC INH ADRES
0720: 0136 A5 F8 READE LDA ALS HET START
0730: 0138 CD 5B 01 CMP EAL ADRES GELIJK IS
0740: 013B D0 0A BNE READF GEWORDEN AAN
0750: 013D A5 F9 LDA INH HET EIND ADRES+1
0760: 013F CD 5C 01 CMP EAH DAN ZIJN WE KLAAR
0770: 0142 D0 03 BNE READF EN GAAN WE TERUG
0780: 0144 4C 1D 1C JMP RESET NAAR DE RESET ROUTINE
0790: 0147 A9 19 READF LDAIM TWEEH ZET 2 1/2 BIT TIJD
0800: 0149 D0 CB BNE READB
0810: 014B 8D F7 1A READG STA 1K TIMER TIJD
0820: 014E 2C F7 1A READH BIT LAAT TIMER
0830: 0151 10 FB BPL AFTELLEN
0840: 0153 AD 82 1A READI LDA LEES POORT B (PB6)
0850: 0156 29 40 ANDIM #40 ZET ONGEMENSTE BITS UIT
0860: 0158 60 RTS TERUG NAAR HOOFD ROUTINE
0870: 0159 00 SAL START ADRES LOW
0880: 015A 00 SAH START ADRES HIGH
0890: 015B 00 EAL EIND ADRES LOW (EIND ADRES+1)
0900: 015C 00 EAH EIND ADRES HIGH

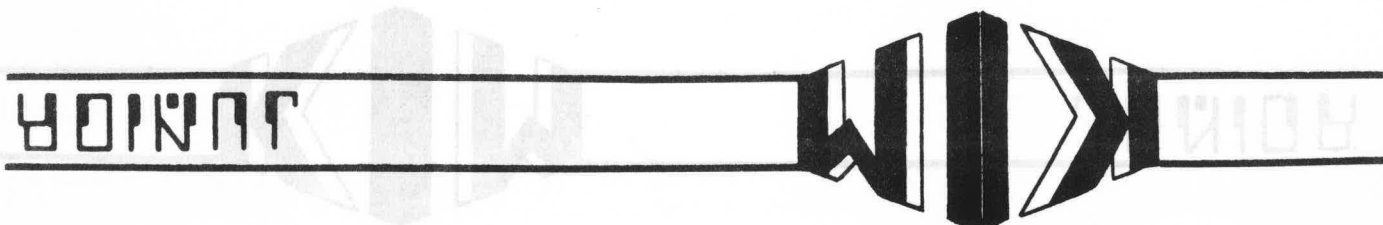
```

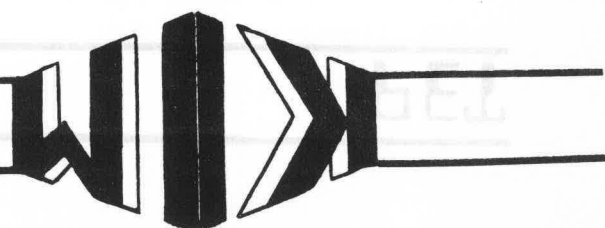
SYMBOL TABLE 3800 388A

```

0010: 1A00 WRITE ORG $1A00
0020:
0030: *****
0040: *
0050: * SCHRIJF ROUTINE VOOR KANSAS CITY CASSETTE
0060: *
0070: * INTERFACE VOOR DE JUNIOR MICRO COMPUTER
0080: *
0090: * POOTJE ZO VAN DE APPLICATIE CONNECTOR
0100: *
0110: * DOORVERBINDEN MET DE CASSETTE OUTPUT
0120: *
0130: * START ADRES INLIKEN OP SAL/SAH
0140: *
0150: * EIND ADRES + 1 INLIKEN OP EAL/EAH
0160: *
0170: *
0180: *
0190: *
0200: *
0210: *
0220: * HALF * $0005 (HALVE BIT TIJD = 5)
0230: *
0240: * HEEL * $000A (HELE BIT TIJD = 10)
0250: *
0260: * TWEEH * $0019 (TWEE EN EEN HALVE BIT TIJD = 25)
0270:
0280:
0290:
0300:
0310:
0320:
0330:
0340:
0350:
0360:
0370:
0380:
0390:
0400:
0410:
0420:
0430:
0440: 1A00 AD 56 1A LDA SAL
0450: 1A03 85 F8 STA INL
0460: 1A05 AD 57 1A LDA SAH
0470: 1A08 85 F9 STA INH
0480: 1A0A A9 20 LDMAI $20
0490: 1A0C 8D 83 1A STA PBD
0500: 1A0F A0 00 WRITEA LDVIM $00

```





0510:	1A11 B1 F8	LDAI INL	HAAL EEN BYTE OP
0520:	1A13 20 2D 1A	JSR	WRITEC EN SCHRYF HEM WEG
0530:	1A16 E6 F8	INC	VERHOOG
0540:	1A18 D0 02	BNE	WRITEB START ADRES
0550:	1A1A E6 F9	INC	MEET 1
0560:	1A1C A5 F8	WRITEB LDA	ALS HET START
0570:	1A1E CD 58 1A	CMP	ADRES GELIJK IS
0580:	1A21 D0 EC	BNE	WRITEA GEMORDEN AAN
0590:	1A23 A5 F9	LDA	HET EIND ADRES +1
0600:	1A25 CD 59 1A	CMP	DAN ZIJN WE KLAAR
0610:	1A28 D0 E5	BNE	WRITEA EN GAAN WE TERUG
0620:	1A2A 4C 1D 1C	JMP	NAAR DE RESET ROUTINE
0630:	1A2D 18	WRITEC CLC	STOP DE TE SCHRYVEN
0640:	1A2E 48	PHA	BYTE OP DE STACK
0650:	1A2F 20 43 1A	JSR	SCHRYF START BIT WEG
0660:	1A32 68	PLA	HAAL TE SCHRYVEN BYTE OP
0670:	1A33 A0 08	LDAI INL	ZET AANTAL BITS PER BYTE
0680:	1A35 0A	WRITEC ASLA	SCHUIF EEN BIT IN DE CARRY FLAG
0690:	1A36 48	PHA	ZET HET RESTANT OP DE STACK
0700:	1A37 20 43 1A	JSR	SCHRYF EEN BIT WEG
0710:	1A3A 68	PLA	HAAL RESTANT TERUG VAN DE STACK
0720:	1A3B 88	DEY	HERHAAL DIT TOTDAT WE 8 BITS
0730:	1A3C D0 F7	BNE	HEBBEN WEGGESCHREVEN
0740:	1A3E 38	SEC	SCHRYF TWEE
0750:	1A3F 20 43 1A	JSR	WRITEC STOP BITS
0760:	1A42 38	SEC	WEG
0770:	1A43 A2 00	WRITEC LDIXM \$00	SCHRYF EEN
0780:	1A45 90 01	BCC	WRITEC BIT NAAR
0790:	1A47 CA	DEX	P1A POORT B
0800:	1A48 8E 82 1A	WRITEC STX	PBD BIT 5 (PBD)
0810:	1A4B A9 0A	LDAI INL	HEEL ZET HELE BIT TIJD
0820:	1A4D 8D F7 1A	STA	CNTD VOOR TIMER
0830:	1A50 2C F7 1A	WRITEC BIT	EN LAAT TIMER
0840:	1A53 10 FB	BPL	WRITEC AFLOPEN
0850:	1A55 60	RIS	GA TERUG NAAR HOOFD ROUTINE
0860:	1A56 00	SAL	=
0870:	1A57 00	SAH	=
0880:	1A58 00	EAL	=
0890:	1A59 00	EAH	=

SYMBOL TABLE 3800 387E

CNTD	1AF7	EAH	1A59	EAL	1A58	HALF	0005
HEEL	000A	INH	00F9	INL	00F8	PBDD	1A83
PBD	1A82	RESET	1C1D	SAH	1A57	SAL	1A56
TWEH	0019	WRITE	1A00	WRITEA	1A0F	WRITEB	1A1C
WRITEC	1A2D	WRITED	1A35	WRITEE	1A43	WRITEF	1A48
WRITEG	1A50						

De te gebruiken USER PORT bestaat uit een 8-bits databus welke is aangesloten op de data A uitgang van een 6522 VIA (Output Register A) Het genoemde ORA wordt bij een reset van de CBM "hardware" geïnitia- liseerd, dus door het laag worden van de reset van het VIA. Het ORA heeft adres \$E84F en bevat dan \$FF. Het is rechtstreeks geschikt voor input "active low", zolang het eveneens door de reset op 0 ge- initialiseerde DATA DIRECTION REGISTER A (DDRA) in deze toestand blijft. Het DDRA heeft in de CBM \$E843 als adres. Door een bit van het DDRA 1 te maken zal het overeenkomstig bit van het ORA nul worden en geschikt zijn voor output (active high) Eenmaal op 1 gezet zal dat bit dan toch weer als active-low input kunnen dienen, zonder dat men iets aan de inhoud van DDRA behoeft te doen.

## DE CBM USERPORT

In de +6V-leiding van de motor is een weerstand van 2,2 ohm opgenomen. Als de band is teruggespoeld en dus stil staat zal de motor, die nu het slippend mechanisme moet aandrijven, veel meer stroom gaan opnemen zodat de BC-557 opent. Een eenvoudige filter stelt de gevoeligheid in en filtert de "troep", veroorzaakt door de draaiende motor uit. Ook voorziet het in enige vertraging, zodat niet elke stroompiek een output naar pen E van de USERPORT stuurt, op welk punt de CBM kan lezen of een bandspooloperatie is afgelopen.

Op pen D van de USERPORT kan de CBM nu geheel onafhankelijk zien of er een recordertoets is ingedrukt. De lijn "sense" die de CBM vertelt of een recordertoets is ingedrukt wordt onderbroken en via de dubbele transistor-inverter weer aangeslo- ten. In plaats van te worden geaard door de recorderswitch, gebeurt dat nu door het met deze schakelaar afknijpen van de eerste tor, zo- dat de tweede opent. Het effect is dus voor de CBM gelijk, maar we kunnen nu zonder de schakelaar te openen de sense-lijn hoog maken, waardoor de motor stopt, door pen C van de USERPORT door de CBM zelf

Altereerst is het nodig de cassette recorder geschikt te maken voor ons doel. De schakeling van fig 1 past gemakkelijk op een stukje print of montaprint van 2\*4 cm.

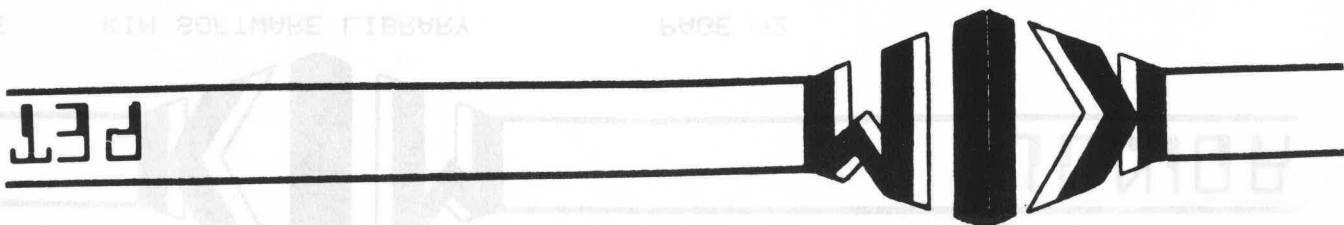
2 Een klein BASIC programma in gebruik! Zonde van de lege resterende RAM capaciteit. We zullen over meerdere programma's tegelijk in het RAM kunnen gaan beschikken. In mijn voorbeeld twee. Op dit thema mag U zelf eindeloos verder gaan hobbyen.

1. Het ergelijke zoeken naar een file op cassette. We rusten de PET uit met "FAST FORWARD"

Dat is de naam die ik heb gegeven aan een machinetaalprogramma van 192 bytes lang. Precies, op de byte af, passende in de buffer van de niet gebruikte tweede cassette recorder. Het programma staat bijna net zovellig als in rom, als we netjes de in de vorige aflevering besproken voorzieningen hebben aangebracht. Het programma gaat twee onhebbelijkheden van CBM/PET vermoorden:

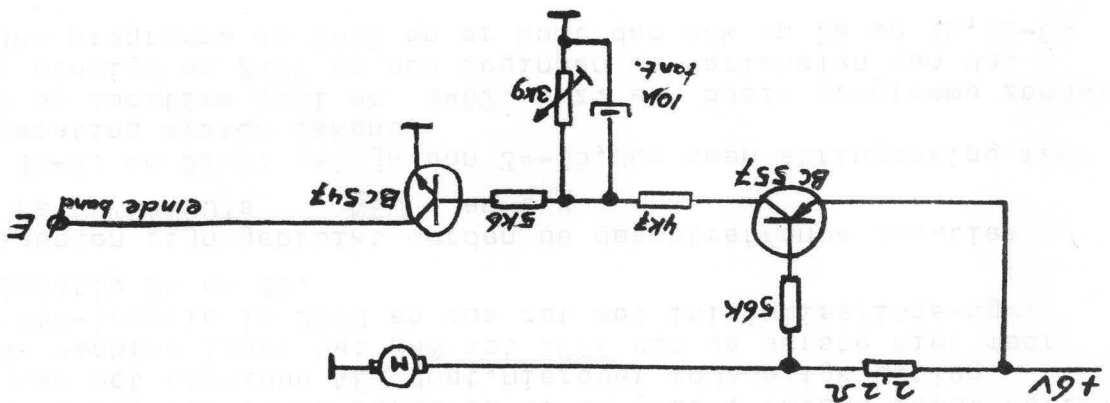
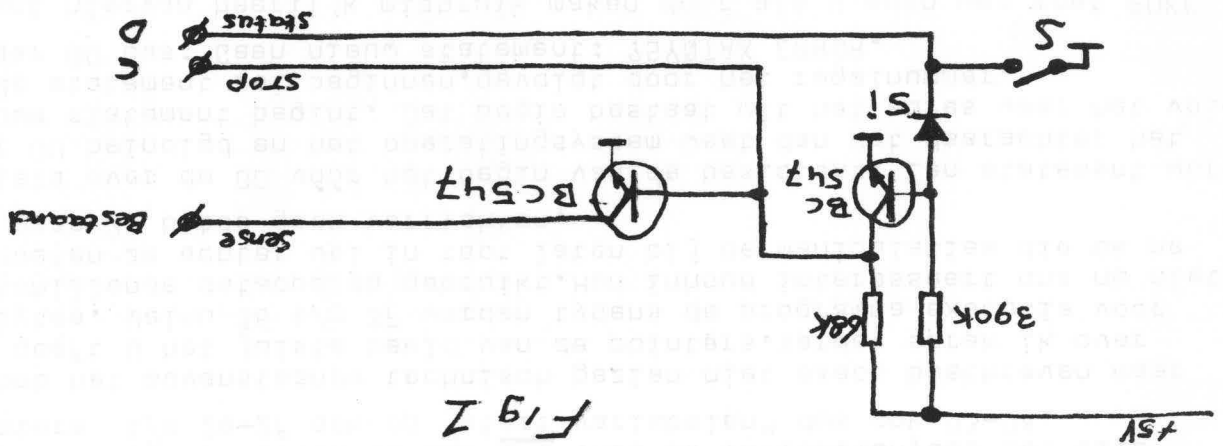
R. Uffhoff

INITIALISATOR





Een machinetaalprogramma in de 2e cassette-buffer zal straks de software gaan verzorgen. We zijn echter nog niet aan dat programma toe omdat het een "link" heeft met een tweede programma in dezelfde buffer dat ons de mogelijkheid geeft ons geheugen in twee delen van ca 4K te verdelen en dan over twee basic-programma's tegeeltijk te beschikken. Gelukkig bezitters van 16 of 32K machines kunnen natuurlijk nog veel verder gaan. Dan wordt het echter verplicht EPROM-programmeeren want de buffer wordt dan te klein.

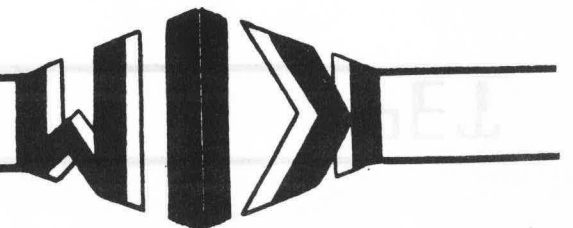


OUTPUTS		adres	\$E843	01	motor stroomloos	•	00	motor kan lopen
FE	geen toets	ingedrukt op recorder						
FA	wel toets	ingedrukt op recorder						
F8	wel toets	ingedrukt op recorder						en einde band bereikt.

In het navolgende zullen we de recordermotor stop zetten door bit 0 van het DDRA op 1 te zetten, zodat bit 0 van het DRA 0 wordt. We kunnen door bit 0 van het DDRA weer nul te maken de motor weer vrij geven.

Laat U straks niet bedriegen door het feit dat dat bit ook bij gesloten DDRA nul blijkt te zijn. De oorzaak kunt U in het schema van fig 1 zien, waar pen C laag wordt gehouden door de basis-emitter overgang van de laatste BC 547. Het bit heeft dus "een input" die echter van geen betekenis is voor ons doel.

INPUTS (bit 0 is dus altijd 0) ADRES \$EE84F



Wat we gaan doen is het operatiesysteem bedruiven. Dat maakt gebruik van een 24-tal locaties op de zeropage: (PET:andere adressen!) Alle adressen zijn hexadecimaal, de dataopslag is in de bekende 10-hi volgorde, zoals in elke abs instructie van de 6502.

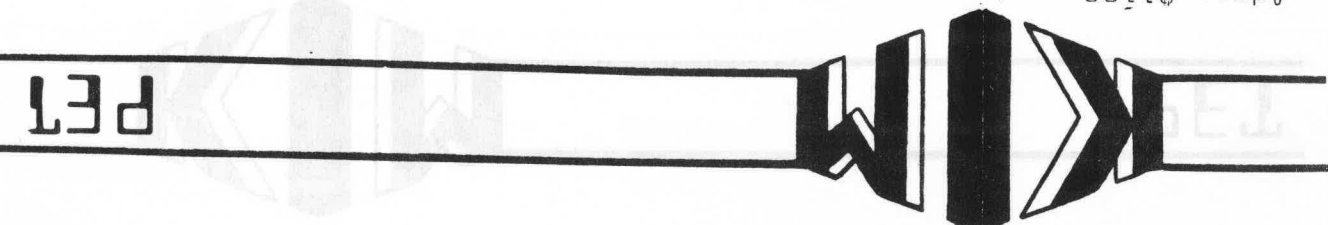
- 28 29 Startadres basistekst.
- 2a 2b Start variabelen opslag.
- 2c 2d Start Array variabelen.
- 2e 2f Eerste vrije RAM locatie
- 30 31 Eerste niet meer vrije RAM locatie, gebruikt voor STRING opslag
- 32 33 Eerste locatie niet meer gebruikt voor string-opslag
- 34 35 Eerste adres dat buiten het beschikbare RAM valt.

Bij een reset initialiseert de CBM als volgt:  
 Geheugenlocatie 0400 is het eerste RAM adres en daar komt 00 te staan als indicatie dat daar een programma gaat beginnen. Over deze 00 straks nog een leuke tip.  
 Nu wordt aangenomen dat BASIC aan het begin van het RAM gaat beginnen dus de pointer 28-28 wordt op 01 04 gezet. Tevens wordt hetzelfde de gedaan wat het commando NEW doet, hierover zodadelijk uitleg.  
 Voor een 8k machine loopt het RAM tot 1fff dus de eerste niet meer bestaande RAM-locatie is 2000 en dus zet het initialiserings-pgm. 00 20 in locatie 34 en 35.

Zodra variabelen zijn gebruikt worden de desbetreffende locaties gezet. Nu de COMMANDO's NEW en CLR  
 New maakt 30-31 en 32-33 gelijk aan 34-35, dus geen stringopslag meer aan het operatie systeem bekend.  
 New zet 00 op locaties 0401 en 0402. Dvz: een basic programma zonder inhoud dat eindigd op 0402 en dus beginnen de variabelen van dat oneigenlijke programma op 0403 en er komt dan ook op 2a en 2b, 03-04 te staan  
 New doet verder hetzelfde als CLR: wis de variabelen, dus zet alle pointers t/m 2e-2f ook op "start variabelen" dus ook 03-04.  
 Ik heb het bovenstaande technisch gezien niet exact beschreven maar het geeft u het juiste beeld van de pointers. Verder sprak ik over 24 bytes. Welnu 36 t/m 3f worden tijdens de programma executie voor verschillende dataopslag gebruikt. Hun inhoud interesseert ons nu niet. We moeten ze echter wel in tact laten bij de manipulaties die we ne met deze 24 bytes gaan verrichten.

Nu iets over de 00 vóór het begin van de basistekst. Een statement wordt door 00 beëindigd en het operatiesysteem weet dan dat daarachter het nieuwe statement begint. Dat begin bestaat uit het adres waar het volgende statement zal beginnen, gevolgd door het regelnnummer.  
 Zonder 00 dus: Geen nieuw statement: 75NTAX ERROR.

U kunt hiervan heerlijk misbruik maken door als u even weg moet POKE 1024,255 te typen. Runnen van het programma is dan onmogelijk en NEW vernietigt uw programma niet. Met POKE1024,0 is alles weer OK.  
 Door allerlei manipulaties met de besproken pointers kunt u het operatiesysteem helemaal naar uw hand zetten. Alle mogelijkheden hier beschrijven is ondoenlijk. We zullen ons dus beperken tot de tweedeling van het RAM. Pagina \$11 is een mooi begin voor het tweede 4k geheugen en dus zetten we daar eerst 00



Adres \$1100 moet nu ook het einde van de eerste 4K zijn en dus moet in 34/35 dat adres gezet worden. Door nu "new" te typen hebben we ons 8K geheugen tot 4K gemaakt.

Voor de tweede mogelijkheid, de laatste 4K, hebben we alle pointers weer te veranderen en dat is natuurlijk steeds een heidens karwei! Daarom creëren we een tweede 24 byte pointer set, waartoe we de laatste 24 bytes \$11E8 - 11FF van het RAM afpikken.

Ten slotte hebben we dan een machinetaalprogramma nodig dat beide pointer-sets kan omwisselen, al naar gelang we het eerste of het tweede programma willen gebruiken. Met deze gegevens moet U nu het hierna afgedrukte programma kunnen volgen. Enig commentaar zal daarbij wellicht gewenst zijn.

\$033a

Een subroutine die wacht op de juiste input op de USER-PORT "geen toets ingedrukt". Door de subroutine op \$033c aan te roepen kunnen de overige inputs van de USER PORT worden opgewacht.

\$0342

Als bij de programmastart nog een recordertoets is ingedrukt zal het programma naar het nog te bespreken NMI programma gaan en alleen vragen de stoptoets in te drukken. Het programma moet dan opnieuw worden gestart.

Vanuit basic is in direct mode met een USER-commando naar het startadres \$0342 gesprongen. Het gehele pgm is dus een subroutine van basic. De parameter van USER bevat het aantal 1/60 seconden dat we de band willen laten spoelen, in 10-attingpoint notatie, opgeslagen in de FAC op adres 00E5-0064. 00E5 bevat daarbij de exponent.

\$0349

Indien geen recordertoets was ingedrukt neemt de routine aan dat de spoeloperatie mag beginnen en drukt de prompt -> af op het scherm. Dan wordt gewacht op het indrukken van de toets RW of FWD op de recorder. (besproken subroutine)

Als de toetsdruk is uitgevoerd wordt ter bevestiging de prompt \* afgedrukt.

\$0353

De waarde in de floatingaccumulator wordt naar een twee bytes integer op \$61 en 62 geconverteerd. (h1-10)

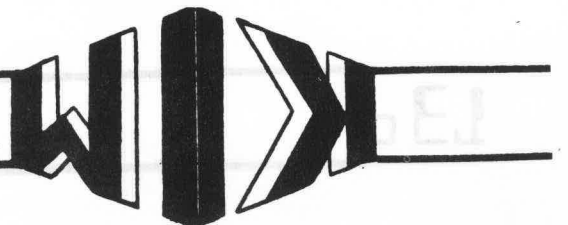
Vervolgens wordt de ingebouwde klok uit de CBM afgelezen. Dit is de starttijd van het programma. We slaan hem op in de bytes \$63 en 64 van de FAC, gewoon omdat dat een even bruikbare zeeopage locatie is. De mantissa's in de FAC zijn toch verder van geen belang meer. Vervolgens wordt de twee bytes-integer (spoeeltijd) van \$61-62 gehaald en met de start-tijd vermeerderd. Het resultaat komt in X en Y register, omdat dat het minste bytes kost. Dit resultaat is de stoptijd!

\$0358

Als we niet een bepaalde tijd voorwaarts wilden spoelen, maar wilden weten hoelang het terugspoelen duurt, is het programma door ons aangeroepen met PRINT USER(0). Nu moet dus bekeken worden of de "spoeeltijd" nul is en dat kan nog steeds, ondanks alle manipulaties met de FAC, door naar de exponent op \$5E te kijken. Nemen we echter eerst aan dat daar geen 0 staat. Het wachten is nu tot de CBM klok gelijk staat aan de waarde in X en Y register. Als dat zover is de juiste tijd voorwaarts gespoeld en moet de band stil gezet gaan worden. Aan-gezien dat ook moet gebeuren naar een REWIND operatie zullen we eerst het geval USER(0) gaan bezien. Daarna gaat het pgm in alle gevallen weer dezelfde weg.

\$036d

\$0369



\$0377

USR was 0. Nu wordt gewacht op het einde van de rewind-operatie, waarbij zich een klein probleem voor doet: De super-snelle processor is al met dit programmadeel bezig als de aanloopstroom van de cassette-motor nog hoog is. Dit zou nu als "einde band" worden opgevat en daarom wordt ca 1/2 sec. gewacht. Pas daarna (weer de subroutine) wordt op ontvangst van het einde-band signaal gewacht.

\$384

Of het einde van de FWD-spoeltijd of eindband nu de reden is doet er niet toe. De motor moet worden gestopt. Eerst wordt hier echter berekend hoe lang de spoel-operatie heeft geduurd. Bij RW is dat het DOEL van het pgm. Bij FWD een leuke controle. Het resultaat van de berekening komt in Y en Accumulator en een subroutine uit de basic-interpretor plaatst het resultaat netjes in floatingspoint in de FAC.

\$390

Gevraagd wordt de stop-toets in te drukken van de recorder en: De motor wordt stil gezet. De enige subroutine laat weer wachten op het indrukken van de stop-toets (=geen toets gedrukt)

\$0395

\$039d

De motor wordt dan weer vrijgegeven en een RTS keert terug naar basic waar de inhoud van de FAC, de spoeltijd wordt afgedrukt.

Om USR te kunnen gebruiken staat in het operatiesysteem op adres \$0000 een constante 4C JMP. Daarnaast moet de gebruiker zelf het juiste adres invullen. Uit pure luiheid laten we dat echter een routine doen op \$03a3, door SYS 931 te typen. Dat hoeft maar een keer, na het laden van het programma te gebeuren en natuurlijk na gebruik van uw zelf aangebrachte resetchakelaar. De routine verandert ook de NMI-vector op \$0094 en 0095, teneinde NMI via ons eigen interrupt-programma naar het normale interrupt-programma op \$C389 te sturen. PET-bezitters moeten dit stukje pgm helaas weglaten omdat hun machine niet over een NMI-RAM-vector beschikt. Het NMI-pgm op 03b2 doet niets anders dan eerst even kijken of er geen cassette I/O operatie bezig is, door het 1sb van de IRQ-vector te controleren. Zoja wordt NMI haastig met een RTI de laan uit gestuurd. Voorts controleert de routine, die we immers veel zullen gebruiken op ons FWD-programma te onderbreken, of we netjes de stop-toets van de recorder hebben ingedrukt. De motor wordt ook terloops even gestopt. Bij elk NMI-gebruik wordt ons tenslotte nog even verteld welk start-adres ons RAM-programma heeft, waaroever meer in dat deel van het tweede programma in de buffer. We zullen nu eerst de gebruiksaanwijzing behandelen, van de FWD faciliteit.

Als U een programma naar cassette gaat schrijven bepaald U en niets of niemand anders voortaan waar op de cassette dat programma komt. Udoet dat als volgt:

PRINTUSR(3740):SAVE"NAAM"

\* 3740

PRESS PLAY & RECORD ON TAPE 1

OK

WRITING NAAM

READY.



Om het programma later terug te vinden typed U:

PRINTUSR(3740):LOAD

de naam mag u gerust weg laten want Uw CBM vindt daar alleen het goede programma. Er blijft nu nog een probleem: Waar mag U het volgen- de programma wegschrijven!

Spoel daartoe na een SAVE de recorder niet terug maar:

PRINTUSR(0)

\* 1731

READY.

Zoek nu in de grafiek van fig 2 de factor V op en vraag:

?1731/V

Noteer het antwoord als plaats voor de volgende file.

U kunt voor elk type cassette het snijpunt M vinden door de totale spoeltijd van de cassette te meten. De grafiek is wat onnauw- keurig omdat alhierhande invloeden zoals langzamer lopende motor aan het einde van RLW niet zijn verwerkt. Deze fouten moet men laten voor wat ze zijn want ze vormen toevallig niet de gewenste veiligheidsmarge.

Het tweede deel van het programma geeft U de mogelijkheid met SYS959 van RAM te verwijsselen:onderste of bovenste 4K. Het programma moet na wat ik over de pointers verteld heb voor zich spreken. Vergaet niet na de eerste SYS959 het commando NLW te typen. Een volgende keer zal ik desgewinst nog een aantal mogelijkheden met deze pointers onder de loop nemen, die te maken hebben met programma OVERLAY en machinetaalroutines in het RAM naast het bijbehorend basic.

Voorts is nog een SOFTWARE blok golf generator in de maak met zeer grote frequentienauwkeurigheid.

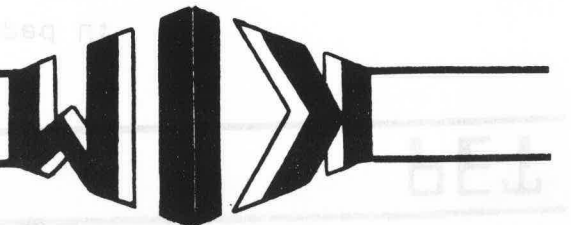
PROGRAMMA

033a a9 fe lda imm  
033c cd 4f e8 cmp abs  
033f d0 fb bne rel  
0341 60 bne rel  
0342 a9 fe lda imm  
0344 cd 4f e8 cmp abs  
0347 d0 70 bne rel  
0349 a9 5f lda imm  
034b 20 32 f2 jsr abs  
034e a9 fa lda imm  
0350 20 3c 03 jsr abs  
0353 a9 2a lda imm  
0355 20 32 f2 jsr abs  
0358 20 9a d0 jsr abs  
035b a5 8f lda zpg  
035d 85 64 sta zpg  
035f 65 62 adc zpg  
0361 a8 tay imp  
0362 a5 8e lda zpg  
0364 85 63 sta zpg  
0366 65 61 adc zpg  
0368 aa tax imp  
0369 a5 5e lda zpg

cont.stoptoets  
bne 03b9 breek pgm af.  
print prompt  
jsr wrt (basic int.)  
jsr flp-int (basic int.)  
jsr clock  
sta lsb clock starttijd  
sta lsb stoptijd  
sta in y lsb stoptyd  
lda msb als boven voor msb

cont op 0 in FAC  
msb sta in X  
lda zpg

036b f0 0a	beq rel	indien 0 rew, zoniet fwd.	
036d e4 8e	cpz zpg	wacht op einde spoeltyd.	
036f d0 fc	bne rel	bne 036d	
0371 c4 8f	cpy zpg		
0373 d0 fc	bne rel	bne 0371	
0375 f0 0d	beq rel	beq 0384	
0377 a5 64	lda zpg	wacht op einde bandsignaal	
0379 69 20	adc lmm	wacht op einde aanloopstroom motor	
037b c5 8f	cmp zpg	bne 037b	
037d d0 fc	bne rel		
037f a9 f8	lda <del>zpg</del>		
0381 20 3c	jsr abs	jsr cont. wacht op einde bandsignaal.	
0384 a5 8f	lda zpg	bereken looptijd band	
0386 e5 64	sbc zpg		
0388 a8 8e	tay lmp		
0389 a5 8e	lda zpg		
038b e5 63	sbc zpg		
038d 20 6d	jsr abs	jsr int-flp (basic int.)	
0390 a9 a6	lda lmm	print prompt	
0392 20 32	jsr abs	jsr wrt (basic int.)	
0395 a9 01	lda lmm	blokkeer motor (stop)	
0397 8d 43	sta abs	jsr cont. wacht op stopstoets	
039d a9 00	lda lmm	geef motor vrij	
039f 8d 43	sta abs	Terug naar basic/of einde srt.	
03a3 a9 42	lda lmm	initialiseer NMI en USR (SYS 931)	
03a5 85 01	sta zpg	adres usr	
03a7 a9 03	lda lmm	page usr	
03ab 85 95	sta zpg	page nmi	
03ad a9 b2	lda lmm	adres nmi (vector 0094/0095 geseet)	
03af 85 94	sta zpg	rtz lmp	
03b1 60	rtz lmp	terug naar basic	
03b2 a9 2e	lda lmm	"masked" nmi programma	
03b4 c5 90	cmp zpg	compare irq-vector: leesoperatie gaande ?	
03b6 f0 01	beq rel	beq 0309	
03b8 40	RTI lmp	leesoperatie: "ignore" NMI	
03b9 20 90	jsr abs	Breek lopend spoelprogramma/ingedrukte toets-af!	
03bc 4c f2	jmp abs	print geheugen(RAM) status	
03bf ad 00	lda abs	memory control program.: Controleer status.	
03c2 f0 1b	beq rel	beq 03df Initialiseer niet onnodig.(fout)	
03c4 a9 00	lda lmm	initialiseer geheugendeeling	
03c6 8d 00	sta abs	00 pointer voor tweede RAM helft	
03c9 a9 01	lda lmm	creëer tweede pointer serie,	
03cb 8d e8	sta abs	en:	
03ce a9 1f	lda lmm	wijzig datapointers.	
03d0 85 35	sta zpg		
03d2 8d e9	sta abs		
03d5 a9 e8	lda lmm		
03d7 8d f4	sta abs		
03da a9 1f	lda lmm		
03dc 8d f5	sta abs		

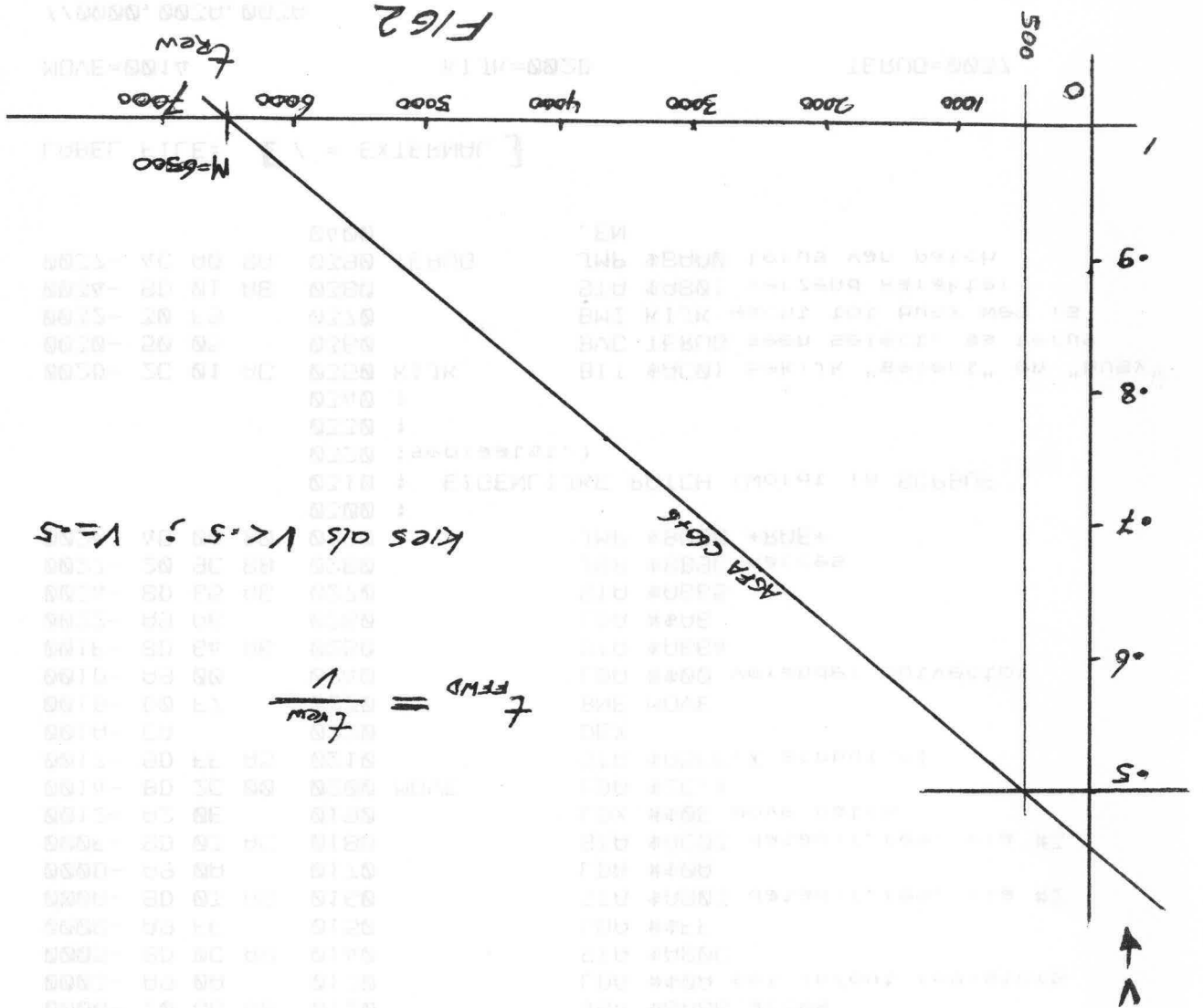


```

03df a2 00 ldx lmm wissel geheugen gebruik (sys 959)
03e1 b5 28 ldx zpx
03e3 48 pha lmm
03e4 bd e8 1f ldx abx
03e7 95 28 sta zpx
03e9 68 pla lmm
03ea 9d e8 1f sta abx
03ed e8 ldx lmm
03ee e0 19 cpx lmm
03f0 d0 ef bne rel
03f2 a5 29 ldx zpg
03f4 20 75 jsr abs
03f7 4c 89 c3 jmp abs

```

warme start basic.  
 jsr wroob (basic int.) "type a byte"  
 print geheugen keuze(status)  
 bne 03el



F/62

```

0010 ;***: RAE PRINTER BOOTSTRAP ***:
0020 ;
0030 ;Laden via normale SYM L2 routine,
0040 ;vervoers G 0 indrukken, waarna men
0050 ;direct in de koude start van de RAE
0060 ;uitkomt.
0080 ;
0090 ;
0100 ;BA 0000
0110 ;OS
0120 JSR $B8B6 acces
0130 LDA #$0A set in/out registers
0140 STA $A80C
0150 LDA #$FF
0160 STA $A803 data dir. reg. via #2
0170 LDA #$0A
0180 STA $AC03 data dir. reg. via #3
0190 LDH #$0E move patch
0200 LDA $2C,X
0210 STA $A5FF,X scbutf -1
0220 DEX
0230 BNE MOVE
0240 LDA #$00 verander outvector
0250 STA $A6E4
0260 LDA #$A6
0270 STA $A6E5
0280 JSR $B89C naces
0290 JMP $B800 *RAE*
0300 ;
0310 ; EIGENLIJKE PATCH (Wordt in SCBUTF
0320 ;geplaatst.)
0330 ;
0340 ;
0350 KIK
0360 BIT $AC01 bekijk "select" en "busy"
0370 BVC TERUG geen select, ga terug
0380 BMI KIK wacht tot busy weg is
0390 STA $A801 verzend karakter
0400 JMP $B8A0 terug van patch
0410 ;
0420 ;
0430 ;
0440 ;
0450 ;
0460 ;
0470 ;
0480 ;
0490 ;
0500 ;
0510 ;
0520 ;
0530 ;
0540 ;
0550 ;
0560 ;
0570 ;
0580 ;
0590 ;
0600 ;
0610 ;
0620 ;
0630 ;
0640 ;
0650 ;
0660 ;
0670 ;
0680 ;
0690 ;
0700 ;
0710 ;
0720 ;
0730 ;
0740 ;
0750 ;
0760 ;
0770 ;
0780 ;
0790 ;
0800 ;
0810 ;
0820 ;
0830 ;
0840 ;
0850 ;
0860 ;
0870 ;
0880 ;
0890 ;
0900 ;
0910 ;
0920 ;
0930 ;
0940 ;
0950 ;
0960 ;
0970 ;
0980 ;
0990 ;
1000 ;

```

MOVE=0014  
 KIK=002D  
 TERUG=0037  
 //0000,003A,003A

LABEL FILE: [ / = EXTERNAL ]



- Ervaringen
- Informatie-uitwisseling
- Verdere ontwikkelingen

Na de publicatie van het KIM-schakprogramma in de KIM-Kenner nr. 11 zijn er een aantal leuke reacties gekomen. Deze reacties hebben geleid tot een aantal vragen over het programma, en een aantal uitbreidingen op het programma. Deze uitbreidingen zijn tot stand gekomen in samenwerking met Joop Tervooren, die veel van het testwerk voor zijn rekening heeft genomen, en die o.m. een vrij uitgebreide tabel met openingen voor het KIM-schakprogramma heeft opgesteld.

1. Het schakprogramma is opgenomen in de KIM-club programma-bibliotheek, zowel in de versie KIM speelt wit als in de versie KIM speelt zwart. Belangstellenden kunnen tijdens de KIM-club bijeenkomsten een copie maken van deze programma's. (Zie regels voor het kopiëren uit de programma-bibliotheek.)

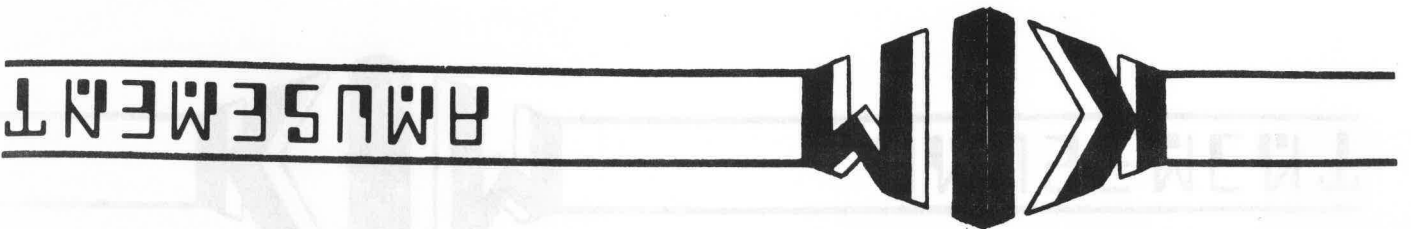
2. Er is enige onduidelijkheid over welke aanpassingen moeten worden gemaakt om de KIM zwart te laten spelen. Voor alle duidelijkheid volgt hier een opsomming van de wijzigingen ten opzichte van het programma in de versie KIM speelt wit:

STY	DSP1	adres	00 10	84 F9
JMP	CODE	"	00 12	4D D4 17
CPY	X'3B'	"	00 C4	CO 3B

Deze wijzigingen moeten worden aangebracht in deel 1 van het programma. (Dus vóórdat deel twee is ingelezen). Hierna kan het gewijzigde deel 1 weer op de band worden gezet (adres 0000-0400); deel twee moet daar ongewijzigd achter worden geplaatst. Wanneer men de wijzigingen vlak voor het spelen met de hand wil inbrengen kan dit nadat beide delen zijn ingelezen. (Maar vóórdat de tweede keer op GO wordt gedrukt.) In dat geval wel eerst de accumulator op nul zetten. (00F3 00), voordat gestart wordt op 0000.

3. Op de volgende pagina's vindt u een openingen-routine en een tabel met openingen voor het KIM-schakprogramma. Dit is uitgewerkt en getest door Joop Tervooren. (Er is wel meer dan 1 K geheugen voor nodig.)

4. Op de daaropvolgende pagina een stukje programma waarmee het schakprogramma snel kan worden geïnitieerd. Door dit stukje programma in te lezen, nadat een partij is gespeeld, kan snel met een nieuwe partij worden begonnen. (Ontwikkeld door Joop Tervooren)



5. Op de pagina daaropvolgend een routine die het schaak- programma versnelt. Ook hier is meer geheugenruimte voor nodig. Deze routine is nog niet geheel uitgetest. Wie probeert dit verder uit ?
6. Op- en aan-merkingen graag aan Theo Kortekaas tel. 02977-21888

DATA /Adressen velden KIM- schaakprogramma T.K.

Veld-  
aanduiding  
schaakbord.

8	4e	4f	50	51	52	53	54	55
7	46	47	48	49	4a	4b	4c	4d
6	3e	3f	40	41	42	43	44	45
5	36	37	38	39	3a	3b	3c	3d
4	2e	2f	30	31	32	33	34	35
3	26	27	28	29	2a	2b	2c	2d
2	1e	1f	20	21	22	23	24	25
1	16	17	18	19	1a	1b	1c	1d

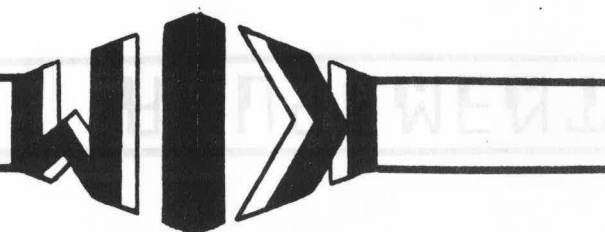
stukken-code  
geplaatst op  
schaakbord::

A	B	C	D	E	F	G	H
84	83	85	86	82	85	83	84
80	80	80	80	80	80	80	80

00 = leeg veld::

A	B	C	D	E	F	G	H
01	01	01	01	01	01	01	01
04	03	05	06	02	05	03	04

Stukken opzetten tijdens het spel::: Problemen??  
RS-AD- veldadres - DA- stukcode - AD - en op beschreven  
manier zet invoeren.



Aanvulling schakelprogramma "Theo Kortekas", KIM-kenner 11, voor diegene met meer dan 1K-geheugenruimte.

Subroutine die het mogelijk maakt openingen in te voeren, welke door de KIM worden aangeroepen, willekeurig en/of kunnen worden aangewezen door de speler, middels een aanwijstabel die men zelf kan vullen.

Tab1 = voor in code bepaalde "afvelden" de vijfde pagina in geheugen.  
 Tab2 = " " "toevelden" 2e pagina=na vorige  
 Tab3 = aanwijstabel voor de zetoelgode 3e pagina=na vorige (openvolgend)  
 zetencode = veldnummering, van links onder naar rechts boven op schakbord,  
 = van 00 tem 3F, of in schakactermen van A1 tem H8.  
 B2-B4, word dan 0C, in tab1: en 1C in tab2: in tab3 komt dan de adressaanduiding,  
 van tab1, waar de volgende zet staat aangegeven, in "LOW Order" aanduiden.  
 0C dus voor 02, wil men alleen deze zet mogelijk maken, dan ipv "0C", 4C aanduiden;  
 In het eerste geval kan de KIM kiezen uit meerdere opeenvolgende zetten, zulke reeksen afsluiten met die ipv., of +4 links in de code.

Patches op 1798 = JSR SRØPEN 20 31 05

DC 05 30 00 (KIM speelt met zwart, dan 01 )

SRØPEN

02

LDA ØTAB1(X)

STA HULP

AND X.BF

CMP ZET3

BNE 04

LDA ØTAB2(X)

CMP ZET2

BNE 04

LDA ØTAB3(X)

BEQ 06

TAX

LDA ØTAB1(X)

STA HULP

BIT HULP

BVS +5

CMP TIMER

BMI 05

AND X.BF

STA ZET3

LDA ØTAB2(X)

STA ZET2

LDA ØTAB3(X)

STA PTR

RTS

BIT HULP

BVS 06

INX

BNE 02

INX

BNE 03

LDX X.00

STX PTR

JMP CALC

05 7C

31 30 05

34 BD 00 07

37 85 14

39 29 BF

3B C5 02

3D D0 2B

41 AE 30 05

43 BD 00 07

45 85 14

47 29 BF

49 C5 01

4B D0 24

4D BD 00 09

4F 85 14

51 24 14

53 70 05

55 CD 04 17

57 30 17

59 29 BF

5B 85 02

5D BD 00 08

5F C5 01

61 D0 24

63 BD 00 09

65 85 14

67 24 14

69 70 05

6B CD 04 17

6D 30 17

6F 29 BF

71 85 02

73 BD 00 08

75 C5 01

77 D0 24

79 BD 00 09

7B 85 14

7D 24 14

7F 70 05

81 CD 04 17

83 30 17

85 29 BF

87 85 02

89 BD 00 08

8B C5 01

8D D0 24

8F BD 00 09

91 85 14

93 24 14

95 70 05

97 CD 04 17

99 30 17

9B 29 BF

9D 85 02

9F BD 00 08

A1 C5 01

A3 D0 24

A5 BD 00 09

A7 85 14

A9 24 14

AB 70 05

AD CD 04 17

AF 30 17

B1 29 BF

B3 85 02

B5 BD 00 08

B7 C5 01

B9 D0 24

BB BD 00 09

BD 85 14

BF 24 14

C1 70 05

C3 CD 04 17

C5 30 17

C7 29 BF

C9 85 02

CB BD 00 08

CD C5 01

CE D0 24

CF BD 00 09

D1 85 14

D3 24 14

D5 70 05

D7 CD 04 17

D9 30 17

DB 29 BF

DD 85 02

DE BD 00 08

DF C5 01

E1 D0 24

E3 BD 00 09

E5 85 14

E7 24 14

E9 70 05

EB CD 04 17

ED 30 17

EF 29 BF

F1 85 02

F3 BD 00 08

F5 C5 01

F7 D0 24

F9 BD 00 09

FB 85 14

FD 24 14

FF 70 05

01 CD 04 17

03 30 17

05 29 BF

07 85 02

09 BD 00 08

0B C5 01

0D D0 24

0F BD 00 09

11 85 14

13 24 14

15 70 05

17 CD 04 17

19 30 17

1B 29 BF

1D 85 02

1F BD 00 08

21 C5 01

23 D0 24

25 BD 00 09

27 85 14

29 24 14

2B 70 05

2D CD 04 17

2F 30 17

31 29 BF

33 85 02

35 BD 00 08

37 C5 01

39 D0 24

3B BD 00 09

3D 85 14

3F 24 14

41 70 05

43 CD 04 17

45 30 17

47 29 BF

49 85 02

4B BD 00 08

4D C5 01

4F D0 24

51 BD 00 09

53 85 14

55 24 14

57 70 05

59 CD 04 17

5B 30 17

5D 29 BF

5F 85 02

61 BD 00 08

63 C5 01

65 D0 24

67 BD 00 09

69 85 14

6B 24 14

6D 70 05

6F CD 04 17

71 30 17

73 29 BF

75 85 02

77 BD 00 08

79 C5 01

7B D0 24

7D BD 00 09

7F 85 14

81 24 14

83 70 05

85 CD 04 17

87 30 17

89 29 BF

8B 85 02

8D BD 00 08

8F C5 01

91 D0 24

93 BD 00 09

95 85 14

97 24 14

99 70 05

9B CD 04 17

9D 30 17

9F 29 BF

A1 85 02

A3 BD 00 08

A5 C5 01

A7 D0 24

A9 BD 00 09

AB 85 14

AD 24 14

AE 70 05

AF CD 04 17

B1 30 17

B3 29 BF

B5 85 02

B7 BD 00 08

B9 C5 01

BB D0 24

BD BD 00 09

BF 85 14

C1 24 14

C3 70 05

C5 CD 04 17

C7 30 17

C9 29 BF

CB 85 02

CD BD 00 08

CE C5 01

CF D0 24

D1 BD 00 09

D3 85 14

D5 24 14

D7 70 05

D9 CD 04 17

DB 30 17

DD 29 BF

DE 85 02

DF BD 00 08

E1 C5 01

E3 D0 24

E5 BD 00 09

E7 85 14

E9 24 14

EB 70 05

ED CD 04 17

EF 30 17

F1 29 BF

F3 85 02

F5 BD 00 08

F7 C5 01

F9 D0 24

FB BD 00 09

FD 85 14

FE 24 14

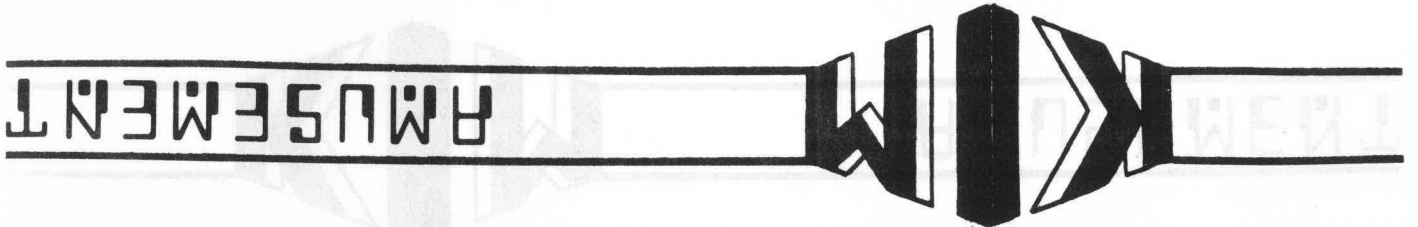
FF 70 05

01 CD 04 17

03 30 17

05 29 BF

0



Om KIM met zwart te laten spelen moet 66k worden gewijzigd  
0027 EA in 0027 00, alvorens te initialiseren

0700) Dit alles kan overal elders in het geheugen,  
0800) mits aanpassingen voor de plaatsbepaling.  
0900)

zetten adres TAB1 TAB2 TAB3  
zetten adres TAB1 TAB2 TAB3

01	b4xc3	2D	59	12	30
05	d7-d5	2E	73	23	31
60	c6-a5	2F	6A	20	32
98	d4-d5	30	5B	23	33
D1	e4xd5	31	5C	23	34
08	c4-d3	32	5A	13	35
3C	c3-f6	33	52	2D	36
4e	f6xd5	34	6D	23	37
0A	d7-d5	35	73	23	38
0B	f1-e1	36	45	04	39
0C	f3xe5	37	55	24	3A
1B	e4-e5	38	5C	24	3B
1A	c6-e7	39	6A	34	00

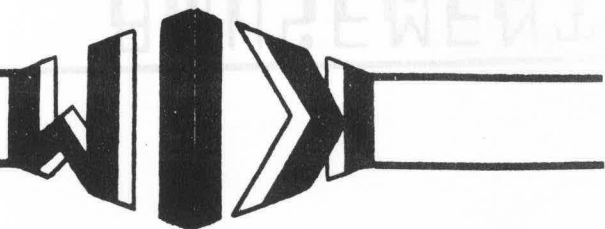
00	e2-e4	1	0C	00	45
01	1C	05	1B	0B	4A
60	d7-d5	60	1A	0A	44
98	d4-d5	98	D1	19	49
D1	e4xd5	31	5C	22	32
08	c4-d3	32	5A	13	35
3C	c3-f6	33	52	2D	36
4e	f6xd5	34	6D	23	37
0A	d7-d5	35	73	23	38
0B	f1-e1	36	45	04	39
0C	f3xe5	37	55	24	3A
1B	e4-e5	38	5C	24	3B
1A	c6-e7	39	6A	34	00

Code Label / opheningen

8	38	39	3A	3B	3C	3D	3E	3F
7	30	31	32	33	34	35	36	37
6	28	29	2A	2B	2C	2D	2E	2F
5	20	21	22	23	24	25	26	27
4	18	19	1A	1B	1C	1D	1E	1F
3	10	11	12	13	14	15	16	17
2	08	09	0A	0B	0C	0D	0E	0F
1	00	01	02	03	04	05	06	07

A B C D E F G H

0 1

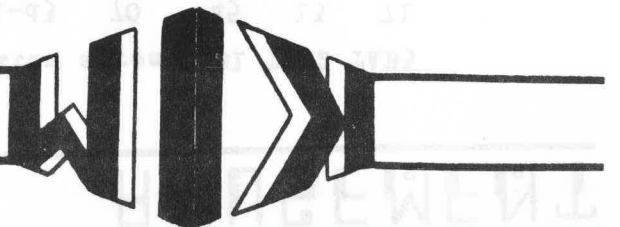


zetten adres TAB1 TAB2 TAB3

zetten adres TAB1 TAB2 TAB3

71	13	45	f1-d3	70
72	23	73	d7-d5	71
73	15	46	o1-f3	72
74	22	72	c7-c5	73
75	06	44	e1-o1	74
76	1A	63	d5xc4	75
77	1A	53	d3xc4	76
78	33	79	b8-d7	77
79	0C	43	d1-e2	78
7A	29	71	b7-b6	79
7B	23	58	d4-d5	7A
7C	12	59	b4xc3	7B
7D	15	46	o1-f3	7C
7E	22	72	c7-c5	7D
7F	23	5A	c4xd5	7E
80	1B	62	c5xd4	7F
81	1B	43	d1xd4	80
82	23	66	e6xd5	81
83	1C	4C	e2-e4	82
84	2A	79	b8-c6	83
85	21	45	f1-b5	84
86	1C	6D	f6xe4	85
87	06	44	e1-g1	86
88	2D	5C	e4-f6	87
89	26	42	c1-o5	88
8A	7D	45	f1-d3	8A
8B	13	74	e7-e6	8B
8C	2C	53	d3xf5	8C
8D	25	6C	e6xf5	8D
8E	25	43	d1-d3	8E
8F	13	7B	d8-c8	8F
90	3A	49	b2-b3	90
91	11	79	b8-a6	91
92	06	44	e1-o1	92
93	34	7D	f8-e7	93
94	1A	4A	c2-c4	94
95	3E	7C	e8-o8	95
96	12	41	b1-c3	96
97	2A	72	c7-c6	97
98	2D	3E	08-f6	98
99	32	74	e7-e5	99
9A	74	41	b1-c3	9A
9B	12	74	e7-e5	9B
9C	24	46	o1-f3	9C
9D	15	79	b8-c6	9D
9E	2A	4E	g2-g3	9E
9F	16	7D	f8-b4	9F
A0	19	52	c3-d5	A0
A1	23	59	b4-c5	A1
A2	22	45	f1-o2	A2
A3	0E	73	d7-d6	A3
A4	2B	06	e1-o1	A4
A5	23	45	f6xd5	A5
A6	06	44	f1-d3	A6
A7	23	45	f6xd5	A7

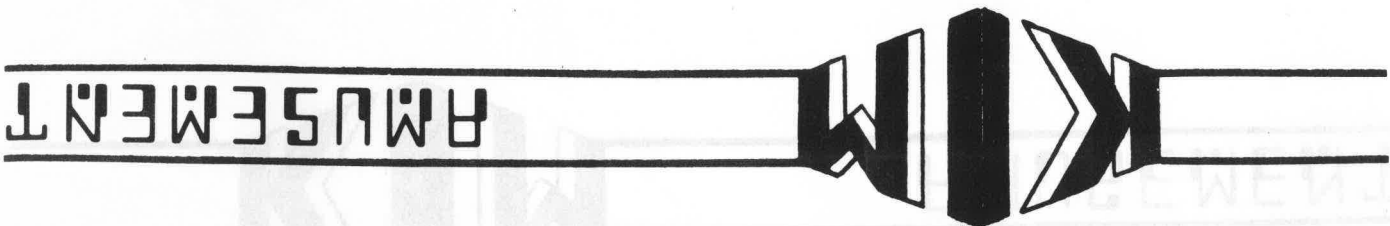




zetten adres TAB1 TAB2 TAB3

[illegible]



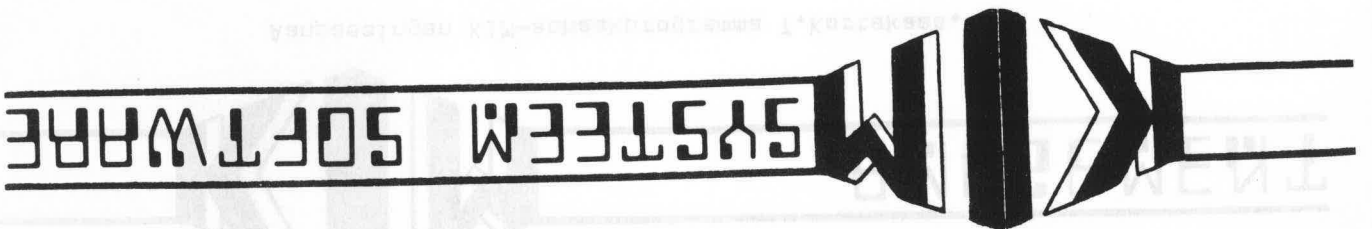


Programma-deel, schakprogramma TK, op de band zetten om snel een nieuwe partij te kunnen gaan spelen.

INITIALISERING "WIT" KIM "ZWART"

INIT1	JMP	INIT2	0000	4C	26	00
INIT5	LDI	X'00'	03	A0	00	
INIT6	LDX	X'1F'	05	A2	1F	
	STY	BORD+16(X)	07	94	26	
	DEX		09	CA		
	BPL	INIT6	0A	10	FB	
	STY	ROCO	0C	84	0A	
	STY	WRDE	0E	84	0B	
	STX	WIS	10	86	90	
	JMP	BRK	12	4C	80	17
			15	FF		
		(deel tab.BORD)	16	84	83	85
			1D	80	80	80
			26	A9	EA	
INIT2	LDA	X'EA'	26	A9	EA	
	STY	INIT1	28	85	00	
	LDA	X'85'	2A	A9	85	
	STY	INIT1+1	2C	85	01	
	NOP		2E	EA		
	NOP		2F	EA		
	NOP		30	EA		
	LDX	X'05'	31	A2	05	
INIT4	LDA	STOP(X)	33	B5	3E	
	STA	VECT(X)	35	9D	FA	17
	DEX		38	CA		
	BPL	INIT4	39	10	FB	
	JMP	INIT5	3B	4C	03	00
		(deel tab.BORD)	3E	00	00	00
			46	C1	C1	C1
			4E	C4	C3	C5
			56			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			
			DF			
			EF			
			FF			
			0F			
			1F			
			2F			
			3F			
			4F			
			5F			
			6F			
			7F			
			8F			
			9F			
			AF			
			BF			
			CF			





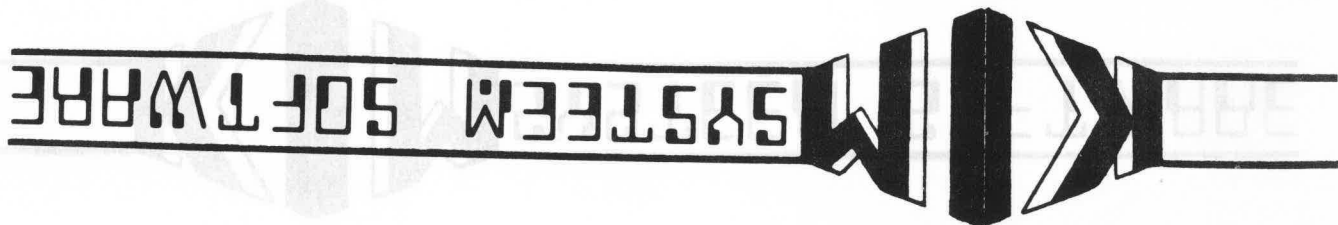
Apple tape's maken met de KIM  
F. Mesander en H.J.C. Otten

Van Frank Mesander uit Leiden kregen we een programma in handen waarmee het mogelijk is om data volgens het Apple formaat te maken, op tape. De software is vrijwel letterlijk gelijk aan die van de Apple monitor, maar dan aangepast aan de KIM omgeving. Voor een andere 6502 machine is deze routine heel makkelijk aan te passen door de lokatie's \$1742 en \$1743 (respectievelijk een PIA en het bijbehorende data direction register) aan de PIA van hun machine aan te passen. Startadres = \$00F1 (1sb) en \$00F2 (msb) Elindadres = \$00F3 (1sb) en \$00F4 (msb) Het signaal naar de cassette recorder is op bit 7 van PIA 1742 te vinden, de gewone audio output van de KIM. Met dit programma is het mogelijk software van de KIM via een cassette naar een Apple over te brengen. Het omgekeerde, van de Apple naar de KIM zou ook handig zijn. Wie schrijft dit voor de club? Andere programma's voor communicatie tussen 6502 computers zijn in de KIM club met zijn vele computer typen zeer welkom.

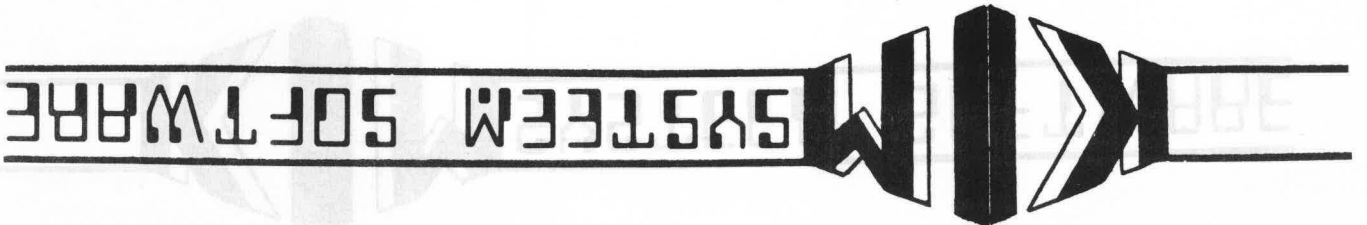
Apple MICRO-MPPE assembler BX-1.0 PAGE 21

0010: 1000	APPLE 090	\$1000
0020: 1000	090	\$1742
0030: 1000	090	\$1742
0040: 1000	090	\$1742
0050: 1000	090	\$1742
0060: 1000	090	\$1742
0070: 1000	090	\$1742
0080: 1000	090	\$1742
0090: 1000	090	\$1742
0100: 1000	090	\$1742
0110: 1000	090	\$1742
0120: 1000	090	\$1742
0130: 1000	090	\$1742
0140: 1000	090	\$1742
0150: 1000	090	\$1742
0160: 1000	090	\$1742
0170: 1000	090	\$1742
0180: 1000	090	\$1742
0190: 1000	090	\$1742
0200: 1000	090	\$1742
0210: 1000	090	\$1742
0220: 1000	090	\$1742
0230: 1000	090	\$1742
0240: 1000	090	\$1742
0250: 1000	090	\$1742
0260: 1000	090	\$1742
0270: 1000	090	\$1742
0280: 1000	090	\$1742
0290: 1000	090	\$1742
0300: 1000	090	\$1742
0310: 1000	090	\$1742
0320: 1000	090	\$1742
0330: 1000	090	\$1742
0340: 1000	090	\$1742
0350: 1000	090	\$1742
0360: 1000	090	\$1742
0370: 1000	090	\$1742
0380: 1000	090	\$1742
0390: 1000	090	\$1742
0400: 1000	090	\$1742
0410: 1000	090	\$1742
0420: 1000	090	\$1742
0430: 1000	090	\$1742
0440: 1000	090	\$1742
0450: 1000	090	\$1742
0460: 1000	090	\$1742
0470: 1000	090	\$1742
0480: 1000	090	\$1742
0490: 1000	090	\$1742
0500: 1000	090	\$1742
0510: 1000	090	\$1742
0520: 1000	090	\$1742
0530: 1000	090	\$1742
0540: 1000	090	\$1742
0550: 1000	090	\$1742
0560: 1000	090	\$1742
0570: 1000	090	\$1742
0580: 1000	090	\$1742
0590: 1000	090	\$1742
0600: 1000	090	\$1742
0610: 1000	090	\$1742
0620: 1000	090	\$1742
0630: 1000	090	\$1742
0640: 1000	090	\$1742
0650: 1000	090	\$1742
0660: 1000	090	\$1742
0670: 1000	090	\$1742
0680: 1000	090	\$1742
0690: 1000	090	\$1742
0700: 1000	090	\$1742
0710: 1000	090	\$1742
0720: 1000	090	\$1742
0730: 1000	090	\$1742
0740: 1000	090	\$1742
0750: 1000	090	\$1742
0760: 1000	090	\$1742
0770: 1000	090	\$1742
0780: 1000	090	\$1742
0790: 1000	090	\$1742
0800: 1000	090	\$1742
0810: 1000	090	\$1742
0820: 1000	090	\$1742
0830: 1000	090	\$1742
0840: 1000	090	\$1742
0850: 1000	090	\$1742
0860: 1000	090	\$1742
0870: 1000	090	\$1742
0880: 1000	090	\$1742
0890: 1000	090	\$1742
0900: 1000	090	\$1742
0910: 1000	090	\$1742
0920: 1000	090	\$1742
0930: 1000	090	\$1742
0940: 1000	090	\$1742
0950: 1000	090	\$1742
0960: 1000	090	\$1742
0970: 1000	090	\$1742
0980: 1000	090	\$1742
0990: 1000	090	\$1742
1000: 1000	090	\$1742





0130:	1000	A5	F1	NXTAO	LDA	AOL
0140:	1002	C5	F3		CMP	ATL
0150:	1004	A5	F2		LDA	AOL
0160:	1006	E5	F4		SBC	ATH
0170:	1008	E6	F1		INC	AOL
0180:	1009	D0	02		BNE	RTSFB
0190:	100C	E6	F2		INC	AOL
0200:	100E	E0			RTSFB	RTS
0210:	100F	A0	4B	HEADR	LDYIM	\$4B
0220:	1011	20	21	10	JSR	ZERDLY
0230:	1014	D0	F9		BNE	HEADR
0240:	1016	E6	F6		ADCM	\$FE
0250:	1018	B0	F3		BCS	HEADR
0260:	1019	A0	21		LDYIM	\$21
0270:	101C	20	21	10	JSR	ZERDLY
0280:	101F	C8			INY	
0290:	1020	C8			INY	
0300:	1021	88			ZERDLY	DEY
0310:	1022	D0	FD		BNE	ZERDLY
0320:	1024	80	05		SBC	WRTPPE
0330:	1026	90	02		LDYIM	\$32
0340:	1028	88			ONEDLY	DEY
0350:	1029	D0	FD		BNE	ONEDLY
0360:	102B	48			WRTPPE	PHQ
0370:	102C	AD	42	17	LDA	PBD
0380:	102F	49	80		EDRIM	\$80
0390:	1031	8D	42	17	STQ	PBD
0400:	1034	68			PLQ	
0410:	1035	90	2C		LDYIM	\$2C
0420:	1037	C8			DEX	
0430:	1038	60			RTS	
0440:	1039	02	08		RDBYTE	LDXIM
0450:	103B	48			RDBYTT	PHQ
0460:	103C	20	47	10	JSR	RDTBIT
0470:	103F	68			PLQ	
0480:	1040	28			ROLQ	
0490:	1041	90	38		LDYIM	\$38
0500:	1043	C8			DEX	
0510:	1044	D0	F5		BNE	RDBYTT
0520:	1046	60			RTS	
0530:	1047	20	48	10	RDTBIT	JSR
0540:	1049	88			RDBIT	DEY
0550:	104B	AD	42	17	LDA	TAPEN
0560:	104E	45	F5		EOR	LASTIN
0570:	1050	10	F8		BPL	RDBIT
0580:	1052	45	F5		EOR	LASTIN
0590:	1054	85	F5		STQ	LASTIN
0600:	1056	C0	80		CPYIM	\$80
0610:	1058	60			RTS	



APPLE MICRO-WARE ASSEMBLER SSXX-1.0 PAGE 03

```
0520: 1055 D9 WRITE CLD
0530: 1058 A9 BF LDAM #BF
0540: 105C 8D 43 17 STA PBDD
0550: 105F A9 27 LDAM #27
0560: 1061 8D 42 17 STA PBD
0570: 1064 A9 40 LDAM #40
0580: 1066 20 0F 10 JSR HEADR
0590: 1069 A0 27 LDYIM #27
0600: 106B 92 00 WRO LDYIM #00
0610: 106D 41 F1 EORL A0L
0620: 106F 48 PHA
0630: 1070 01 F1 LDAX A0L
0640: 1072 20 84 10 JSR WRYTE
0650: 1075 20 00 10 JSR NXTRO
0660: 1078 60 10 LDYIM #10
0670: 107A 60 10 LDYIM #10
0680: 107C 90 EE BCC WRD
0690: 107E 90 22 LDYIM #22
0700: 1080 20 84 10 JSR WRYTE
0710: 1082 F0 47 BEQ BEL
0720: 1084 92 10 WRYTE LDYIM #10
0730: 1086 09 WRYTE
0740: 1087 20 10 JSR WRYTE
0750: 1089 00 F9 BNE WRYTE
0760: 108B 60 RTS
0770: 108D 08 READ CLD
0780: 108F 20 47 10 JSR RDTBIT
0790: 1091 05 15 LDAM #15
0800: 1093 20 0F 10 JSR HEADR
0810: 1095 05 F0 STA CHKSUM
0820: 1098 20 47 10 JSR RDTBIT
0830: 109A 90 24 RDTWO LDYIM #24
0840: 109D 20 4A 10 JSR RDBIT
0850: 109F 80 F9 BCC RDTWO
0860: 10A2 20 4A 10 JSR RDBIT
0870: 10A5 00 2B LDYIM #2B
0880: 10A7 20 39 10 JSR RDTWR
0890: 10A9 01 F1 STIX B0L
0900: 10AB 45 F0 B0R CHKSUM
0910: 10AD 05 F0 STA CHKSUM
0920: 10AF 20 00 10 JSR NXTRO
0930: 10B1 35 LDYIM #35
0940: 10B3 90 F0 BCC RDTWR
0950: 10B5 20 39 10 JSR RDTWR
0960: 10B8 05 F0 CMP CHKSUM
0970: 10BA F0 0D BEQ BEL
0980: 10BC A9 45 LDAM #45
0990: 10BD 20 00 JSR OUTPUT
1000: 10C3 A9 52 LDAM #52
1010: 10C5 20 00 JSR OUTPUT
1020: 10C8 20 00 JSR OUTPUT
1030: 10CB 60 BEL RTS
```

Patches op MICROSOFT BASIC

H.J.C. Otten

Een gemis van Microsoft BASIC ten opzichte van Tiny BASIC (KIM versie) is het ontbreken van berekende regelnummers. Een berekende sprong is vaak erg handig en efficiënt, bovendien kan een subroutine met een naam worden aangeropen wat de duidelijkheid bevordert. Die naam is dan een (integer) variabele die we van te voren een waarde geven. Een voorbeeld:

10 TIOF = 1000

20 A=1 : B=2

30 GOSUB TIOF

40 PRINT C

...

1000 REM SUBROUTINE TIOF , C = A+B

1010 C = A + B

1020 RETURN

Het gebruiken van getallen en niet van expressies heeft het voordeel dat het snel is. De patch moet de snelheid van een GOTO etc niet aantasten liefst. De volgende patch levert het gewenste resultaat:

patch 27F2 in 4C 00 04 ( of een andere lokatie ; )

zet op 0400 : B0 06 BCS REKEN EERSTE KARAKTER EEN GETAL?

20 B828 JSR VERZAMEL REGELNUMMER IN (11,12)

4C F527 JMP verder met routine (normaal)

REKEN 20 342C JSR GET ADDRESS in FPACU

20 EF35 JSR CONVERT ADDRESS TO INTEGER IN (11,12)

4C F527 JMP verder met routine


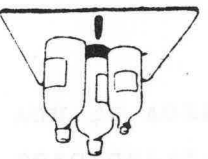
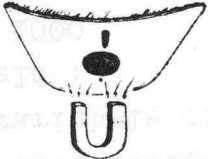

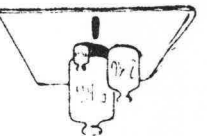
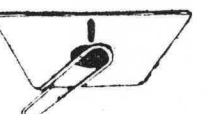
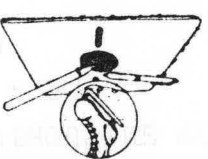
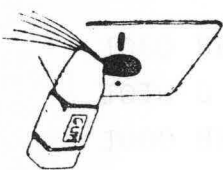

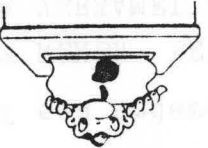
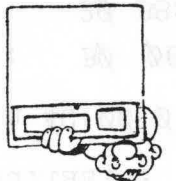
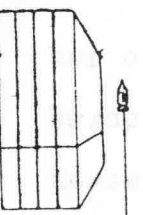
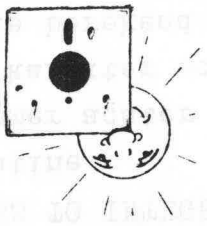
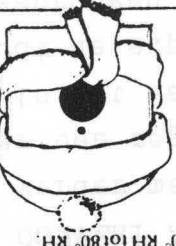
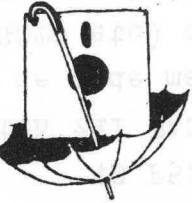
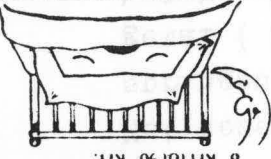
Met deze patch zal BASIC naar een regelnummer achter GOTO etc

springen op de oude manier, is het eerste karakter echter geen

getal ( na GOTO etc ) dan wordt de expressie berekend en naar

het berekende regelnummer gesprongen.

### DE KIM KENNER

<p>1. Raak het oppervlak van de floppy niet met vingers aan. Gevaar op beschadiging, waardoor fouten ontstaan.</p> 	<p>2. Gebruik geen oplosmiddelen zoals alcohol of verdunners om de floppy te reinigen.</p> 	<p>3. Gebruik geen magnetische of statisch geladen voorwerpen vlakbij de floppy. Gegevens op de floppy kunnen verloren gaan, als de floppy aan een magnetisch veld wordt blootgesteld.</p> 	<p>4. De floppy niet buigen of vouwen.</p> 	<p>5. Plaats geen zware voorwerpen boven op de floppy.</p> 	<p>6. Schuif geen klemmen of paper-clips aan de floppy.</p> 	<p>7. Beschrijf het etiket op de floppy nooit met een potlood of balpen. Gebruik een viltstift.</p> 	<p>8. Gebruik geen gum.</p> 	<p>9. Plak het identificatie-etiket op de juiste plaats. Plak nooit etiketten in lagen over elkaar heen.</p> 	<p>10. Floppy aan bovenrand vastpakken en voorzichtig in het aandrijfmecanisme stoppen.</p> 	<p>11. Stop de floppy na gebruik altijd onmiddellijk terug in de bijbehorende enveloppe.</p> 	<p>12. Bewaar de floppy's rechtop in de bijbehorende doos.</p> 	<p>13. Stel de floppy nooit bloot aan overmatige warmte, verhitting of zonlicht.</p> 	<p>14. Omgevingsstemperaturen voor gebruik: 10° C tot 50° C en vochtigheidsgraad: 20% RH tot 80% RH.</p> 	<p>15. Omgevingsstemperaturen voor opslag: 4° C tot 53° C en vochtigheidsgraad: 8% RH tot 80% RH.</p> 	<p>16. Tijdens transport dient de floppy in de bijbehorende enveloppe en een beschermende doos te zijn opgeborgen. Omgevingsstemperatuur: -40° C tot +53° C en vochtigheidsgraad: 8% RH tot 90% RH.</p> 
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------